

XNA Game Studio 3.1

Copyright© 2017 Microsoft Corporation

The content in this document is retired and is no longer updated or supported. Some links might not work. Retired content represents the latest updated version of this content.

XNA Game Studio 3.1

Microsoft XNA Game Studio 3.1 is a set of tools based on supported versions of Microsoft Visual Studio tools that allow students and hobbyists to build games for Microsoft Windows, the Microsoft Xbox 360 video game and entertainment system, and Microsoft Zune. XNA Game Studio also includes the XNA Framework, which is a set of managed libraries based on the Microsoft .NET Framework 2.0 that are designed for game development. This documentation collection contains technology overviews, tutorials, and reference material related to XNA Game Studio.

In This Section

[Frequently Asked Questions and Known Issues](#)

Answers some of the most commonly asked questions about using XNA Game Studio.

[What's New in XNA Game Studio 3.1](#)

Describes new features and changes in Microsoft XNA Game Studio.

[Getting Started with XNA Game Studio](#)

Provides introductory documentation for XNA Game Studio.

[Using XNA Game Studio](#)

Describes how to use XNA Game Studio. XNA Game Studio is seamlessly integrated with supported versions of Microsoft Visual Studio tools, exposing an array of new and updated features for developing 2D and 3D games.

[Programming Guide](#)

Describes how to use the XNA Framework to develop games in XNA Game Studio. The XNA Framework is a set of managed libraries for Windows, the Xbox 360, and Zune. These libraries enable you to be more productive by using a set of unified class libraries to develop C# games.

[XNA Framework Class Library](#)

The XNA Framework class library is a library of classes, interfaces, and value types that are included in XNA Game Studio.

[Content Pipeline Class Library](#)

The Content Pipeline class library is a library of classes, interfaces, and value types that are included in XNA Game Studio.

This library provides access to XNA Framework Content Pipeline functionality and is designed to be the foundation on which Content Pipeline–related applications, components, and controls are built.

[White Papers](#)

Provides technical articles and white papers on XNA Game Studio programming topics.

[XNA Creators Club Online Web Site](#)

More samples and tutorials, as well as developer community forums, are available at the XNA Creators Club Online Web site.

Frequently Asked Questions and Known Issues

Answers some of the most commonly asked questions about using XNA Game Studio.

- [General Questions](#)
- [3D Graphics](#)
- [Input](#)
- [Audio](#)
- [Math](#)
- [Storage](#)

General Questions

Is XNA Game Studio 3.1 Supported on Windows 7 Beta?

XNA Game Studio 3.1 has been tested on Windows 7 Beta. Support for the final release of Windows 7 is expected after it becomes available.

How do I file bugs or make suggestions?

- Sign up for Microsoft Connect at <http://connect.microsoft.com> and click the **XNA Game Studio** connection under available connections.
- After you sign up, you can file a bug or make a suggestion by selecting [Feedback](#) on the **Connect: XNA Game Studio** page.

Is Visual Studio supported with XNA Game Studio?

Yes. XNA Game Studio is designed to install custom extensions to supported versions of Microsoft Visual Studio tools, such as Microsoft Visual Studio 2008 Professional Edition, Microsoft Visual Studio 2008 Standard Edition, and Microsoft Visual C# 2008 Express Edition.

For a complete list of supported editions, see [Microsoft Visual Studio 2008](#) and [Microsoft Visual C# Express Edition](#).

Can I use an earlier version of an XNA Game Studio game project with XNA Game Studio 3.1?

XNA Game Studio 3.1 supports both 3.0 and 3.1 projects, and it includes support for upgrading projects from 3.0 to 3.1.

For more information, please see [Upgrading XNA Game Studio Projects](#).

Can I target Windows, Xbox 360 and/or Zune platforms with the same solution?

Yes. See [Developing Cross-Platform Games](#) for details on how to create and manage games for multiple platforms. This includes information on converting a game project meant for one platform into one to run on another platform, such as a Windows game project into an Xbox 360 console and/or a Zune game project, keeping all projects in the same solution.

Do I need a connection to Xbox LIVE to play my XNA Framework game?

XNA Game Studio Connect requires a connection to Xbox LIVE at all times.

If I have written a multiplayer game, must the player with the XNA Creators Club membership sign in first?

In the event that multiple people wish to play a local multiplayer XNA Framework game, it is essential that the player with a valid XNA Creators Club membership sign in first before launching XNA Game Studio Connect. After launch, additional profiles can then sign in to play.

If multiple profiles on a single Xbox 360 console have a valid XNA Creators Club membership, the connection settings that are used by XNA Game Studio Connect will always default to player one.

How do I distribute my game?

See [Sharing and Distributing Your Game](#).

My computer cannot connect to my Xbox 360 console. What should I do?

This may indicate issues with your network settings, connection key, file names, or file system. Please see [Troubleshooting Xbox 360 Game Deployment](#) for detailed descriptions of these issues.

My computer cannot connect to my Zune device. What should I do?

This may indicate issues with your physical connection, Zune settings or state. Please see [Troubleshooting Zune Game](#)

[Deployment](#) for detailed descriptions of these issues.

What do I do to fix the error message "NoSuitableGraphicsDeviceException was unhandled"?

Check the inner exception for more detail on why the graphics device could not be created. XNA Game Studio thoroughly checks why the graphics device could not be created and reports these checks in the inner exception.

If **NoSuitableGraphicsDeviceException** does not have an inner exception, it is likely that your graphics card does not have a DirectX 9 driver, that it does not support Shader Model 1.1, or that the hardware acceleration slider is not set to **Full** in your settings.

To determine the version of DirectX installed on the computer

1. Click **Start**, and then click **Run**.
2. In the **Open** box, type **dxdiag**, and then click **OK**.

The **System** tab will display information about the version of DirectX that is installed on the computer.

To set hardware acceleration to Full on Windows XP

1. Right-click the desktop, and then click **Properties** on the menu.
2. Click the **Settings** tab, and then click **Advanced**.
3. Click the **Troubleshoot** tab.
4. Move the **Hardware Acceleration** slider until it is set to **Full**.
5. Click **OK**, and then click **Close**.

To set hardware acceleration to Full on Windows Vista

1. Right-click the desktop, and then click **Personalize** on the displayed context menu.
2. Click **Display Settings**
3. Click **Advanced Settings....**
4. Click the **Troubleshoot** tab.
5. Click the **Change Settings** button, then move the **Hardware Acceleration** slider until it is set to **Full**.

Note that for display adapters that do not allow hardware acceleration to be changed, the **Change Settings** button may be disabled.

6. Click **OK** on the three open dialog boxes.

To determine which shader models are supported

Please see [How To: Check for Shader Model 2.0 Support](#) for information about how to programmatically query the shader capabilities of the graphics card.

How do I playback video with XNA Game Studio?

The XNA Framework supports video playback for purposes such as opening splash and logo scenes, cut scenes, or in-game video displays.

For more information, please see [How To: Play Video](#) and [How To: Play a Video in 3D space](#).

Why does my application work from my local computer, but throw SecurityException when I move it to a shared network folder?

This is a security feature of the Common Language Runtime (CLR). Applications that are run from locations other than the local system have greater security restrictions than applications run locally. Running XNA Game Studio titles from a shared network folder is not supported.

What shader models does XNA Game Studio support?

XNA Game Studio supports all the DirectX 9 shader models (versions 1.x, 2.x, and 3.0) on Windows, along with some variants. For a full list of supported shaders, see the [ShaderProfile](#) enumeration.

On Xbox 360, XNA Game Studio supports Shader Model 2.0 and an extended variant of 3.0 customized for the Xbox 360.

Some starter kits may have more restrictive requirements.

Can I use the delay sign feature for my Xbox 360 game?

Use of delay signing can prevent the game from running on the Xbox 360, because the console uses strong name verification. Under strong name verification, the assemblies will not load until the signing process is complete.

When debugging, sometimes my full-screen game crashes and I am unable to return to Visual Studio. What can I do besides rebooting my computer?

When a program takes over the graphics device, sometimes it is unable to properly return the graphics device to Windows after a crash. Try the following:

1. Press the **Windows** button or CTRL+ESC to bring up the **Start** menu.

Sometimes this will restore the task bar, allowing you to bring up Visual Studio.

2. Press ALT+TAB to return focus to Visual Studio, then SHIFT+F5 to stop the program you are debugging.

Sometimes you may have to press ALT+TAB a few times to return the focus to Visual Studio before pressing SHIFT+F5.

3. Press CTRL+ALT+DELETE to attempt to bring up the Windows Security window.

If the Windows Security window comes up, you can use Task Manager to end the task running your game and return to Visual Studio.

If those steps are unsuccessful, the only option might be to reboot the computer.

3D Graphics

How do I create lighting and material effects in XNA Game Studio?

Lights and materials are implemented in XNA Game Studio with effects. To implement lighting or material effects, create an effect that contains the desired vertex or pixel color transformation. The actions of a simple effect are encapsulated in the [BasicEffect](#) class, which provides functionality for applying lights and materials, and setting world, view, and projection transformations. You may choose to use [BasicEffect](#) for simple functionality or create your own effects using the [Effect](#) class.

How do I create my own effects?

For a sample of a simple effect using the [Effect](#) class that sets the diffuse color of a vertex, see [How To: Create and Apply Custom Effects](#).

What is an effect file?

An effect is a combination of vertex and pixel shader code grouped together to encapsulate a particular rendering technique. Effects can be written in either high-level shader language (HLSL) or assembly code.

An effect file contains effect code. You may see effect files with an .fx extension that contain HLSL code. Effects in HLSL or assembly code must be compiled to a binary format before they can be used in an application.

For overviews of HLSL and the effect file format, see the [HLSL Shaders](#) and [Effects](#) DirectX Programming Guides on MSDN. Complete reference documentation for HLSL, shader ASM, and the effect file format is available in the [Direct3D API Reference](#).

After drawing using Graphics.SpriteBatch, why do my 3D objects draw incorrectly?

By default, [SpriteBatch.Begin](#) does not save your current render state, and will change certain render state properties that may make 3D objects render incorrectly. You can choose to either reset the render state yourself after the call to [SpriteBatch.End](#), or call [SpriteBatch.Begin](#) and pass in [SaveStateMode.SaveState](#), which will restore the render state after sprites are drawn.

Can I use PIX for Windows from the DirectX SDK with an XNA framework game?

Yes. PIX sees XNA Framework games as just another DirectX game and will expose information about the DirectX calls made by your XNA Framework game, including shader rendering. To use PIX with an XNA Framework game, start a New Experiment in PIX and enter your game executable path into the Program Path edit box. Then choose the information you want to gather and push Start Experiment.

For more information on how to use PIX, see the DirectX SDK documentation.

Input

Can I map the Xbox Guide button on the Xbox 360 Controller to an action?

No. The **Xbox Guide** button is reserved.

My mouse cursor is not visible. How can I make it visible?

Mouse support is available only for Windows. Set the [Game.IsMouseVisible](#) property to **true**.

The Xbox 360 Controller does not work properly in Windows. How can I correct this?

For the Xbox 360 Controller to work correctly in Windows, you need to install the latest Xbox 360 Controller driver. You can download this driver from [Microsoft Hardware Download Gaming Software](#).

Audio

Can I load and play a wave file from the Audio API without using the XACT tool?

Yes, make a call to the [Play](#) method. For more information, see [How To: Play a Sound](#).

How can I get more information on using XACT?

For an easy guide to adding sound files to an XACT project and playing them, see [How To: Add a Sound File to Your Game Using XACT](#) and [How To: Play a Sound Using XACT](#). For detailed information on how to author audio in the XACT tool, including information on categories, variables, and other advanced features, see [XACT Audio Authoring](#).

Why does calling GetCue from SoundBank never return a cue?

In your XACT project, you may have set a sound bank property called **IncludeCueNames** to **false**. For each sound bank in your project, this property must be set to **true**. Open XACT, load your project, and click each of your sound banks in the project tree view. Check the value of **IncludeCueNames** in the property pane as you click each sound bank. If any of these values are set to **false**, set them to **true** and rebuild your project.

Math

What coordinate system does the XNA Framework use?

The XNA Framework uses a right-handed coordinate system.

Storage

Why do I get an exception when I try to use Storage classes?

One possible reason is that one of your operating system folders — especially %_NTDrive%\Documents and Settings\ — is mapped to a network drive. Storing player data or running titles from a network location is not supported in this release.

Note

For answers to more questions about XNA Game Studio and the XNA Framework, see XNA Creators Club Online .

What's New in XNA Game Studio 3.1

Describes new features and changes in Microsoft XNA Game Studio.

- [New Framework Features](#)
- [New Conceptual Content](#)
- [Changes to the Development Environment and Tools](#)
- [Changes to the XNA Framework API](#)

New Framework Features

- [Avatars](#)
- [Xbox LIVE Party](#)
- [Video](#)
- [Audio Enhancements](#)
- [Content Pipeline Enhancements](#)

Zune Touch-Screen and Accelerometer Support

XNA Game Studio provides support for developing applications that use input gathered from the built-in touch screen and accelerometer of the Zune HD device.

Concepts

[Zune HD Input Overview](#)

Provides an overview of the touch screen and accelerometer input features for the Zune HD device.

Tasks

[Platformer: Adding Touch Support](#)

Extends the base Platformer starter kit code by adding touch screen and accelerometer support for input.

Reference

[Accelerometer Class](#)

Provides methods for interacting with the 3-axis accelerometer of a Zune device.

[AccelerometerCapabilities Structure](#)

Provides properties for accessing the capabilities of an accelerometer.

[AccelerometerState Structure](#)

Provides information on the current state of the accelerometer device and a helper function for rotational computation.

[TouchCollection Structure](#)

Provides methods and properties for accessing state information for the touch screen of a Zune device.

[TouchLocation Structure](#)

Provides methods and properties for interacting with a touch location on a touch screen device.

[TouchPanel Class](#)

Provides methods for retrieving touch panel device information.

[TouchPanelCapabilities Structure](#)

Provides access to information about the touch pad device.

Avatars

Avatars are three-dimensional animated characters. The Xbox Dashboard uses avatars to represent gamers who are signed in to the local console, and players in the gamers' friends lists and LIVE Party chats. Titles can also use avatars in games, both to represent gamers and to represent other characters in the game.

Concepts

[Programming with Avatars](#)

Discusses the support for avatars in XNA Game Studio applications targeting the Xbox 360 console.

Tasks

[How To: Render and Animate an Avatar Using AvatarRenderer.](#)

Demonstrates how to render and animate a gamer's avatar using the [AvatarRenderer](#) class and a standard animation.

Reference

[AvatarAnimation](#)

Provides methods and properties for animating an avatar using standard animations (for example, celebrate).

[AvatarDescription](#)

Provides access to the methods and properties of the description data for an avatar.

[AvatarExpression](#)

Contains the various components of the avatar's face, such as the left and right eyebrows.

[AvatarRenderer](#)

Provides properties and methods for rendering a standard avatar.

[SignedInGamer.AvatarChanged](#)

Occurs when a gamer's avatar changes.

[SignedInGamer.Avatar](#)

Description data for the avatar that represents the gamer.

[AvatarAnimationPreset](#)

Defines standard animations for avatars.

[AvatarBone](#)

Defines a list of the useful bones of the avatar model.

[AvatarEyebrow](#)

Defines the standard animation textures for an avatar's eyebrows.

[AvatarEye](#)

Defines the standard animation textures for an avatar's eyes.

[AvatarMouth](#)

Defines the standard animation textures for an avatar's mouth.

Xbox LIVE Party

Xbox LIVE Party enables gamers to communicate, even when each gamer is not playing the same game in the same multiplayer session. LIVE Party supports up to an eight-way group voice chat for gamers. It does not matter what each gamer is doing on his or her Xbox 360 at the time—playing games, watching videos, listening to music, or browsing the Marketplace. LIVE Party chat keeps gamers connected before, during, and after a gameplay session, persisting across title switches. Furthermore, LIVE Party chat provides easy and quick ways for gamers to get into multiplayer games together.

Tasks

[How To: Add LIVE Party Support](#)

Describes how to add LIVE Party support to Xbox LIVE Indie Games.

Reference

[LocalNetworkGamer.SendPartyInvites](#)

Sends game invitations to all party members that are not in the current game session.

[SignedInGamer.PartySize](#)

Gets the current party size.

[LocalNetworkGamer.SendPartyInvites](#)

Sends game invitations to all party members that are not in the current game session.

[Guide.ShowPartySessions](#)

[Guide.ShowParty](#)

Video

XNA Game Studio now supports the ability to play back video that can be used for such purposes as opening splash and logo scenes, cut scenes, or in-game video displays.

This set of XNA Framework APIs supports the following features:

- Full screen video playback
- Video playback to simple textures in game
- Control of playback such as pause/resume and stop
- Retrieve properties of the video, such as playback time, size, and frame rate
- Determine the type and usage of the audio track, such as if it has music, dialog, or music and dialog
- Play back multiple video streams at the same time

Tasks

[How To: Play Video](#)

Demonstrates how to use the [VideoPlayer](#) to play back videos.

[How To: Play a Video in 3D space](#)

Demonstrates how to use the [VideoPlayer](#) to playback videos on the surface of a quad.

Reference

[Video](#)

Represents a video.

[VideoPlayer](#)

Provides methods and properties to playback, pause, resume, and stop video. [VideoPlayer](#) also exposes repeat, volume, and

play position information.

[VideoSoundtrackType](#)

Type of sounds in a video

Audio Enhancements

This version of XNA Game Studio has a new usage pattern of **SoundEffect.Play**. Sound instances created by **Play** calls are disposed automatically when playback ends, and **SoundEffect.Play** returns a Boolean to indicate success or failure.

Tasks

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Demonstrates how to apply 3D positioning effects to SoundEffects.

[How To: Change the Pitch or Volume of a Sound](#)

Demonstrates how to change pitch and volume of a playing sound.

Content Pipeline Enhancements

This changes the semantics of the Content Pipeline, making it much easier to add custom types.

Reference

[ContentSerializerRuntimeTypeAttribute](#)

A custom [Attribute](#) that specifies the corresponding run-time type of this object.

[ContentSerializerTypeVersionAttribute](#)

A custom [Attribute](#) that specifies the corresponding run-time type version of this object.

[ContentTypeWriter.CanDeserializeIntoExistingObject](#)

Determines if deserialization into an existing object is possible.

New Conceptual Content

Graphics

[The XNA Rendering Pipeline](#)

Provides a high-level view of the graphics rendering pipeline for XNA games.

[What Is a Model Bone?](#)

A model bone is a matrix that represents the position of a mesh relative to other bones in a 3D model.

Changes to the Development Environment and Tools

Visual Studio Changes

XNA Game Studio 3.1 supports both 3.0 and 3.1 projects, and it includes support for upgrading projects from 3.0 to 3.1.

Tasks

[Upgrading XNA Game Studio Projects](#)

Describes supported and unsupported XNA Game Studio project versions, provides guides for upgrade scenarios, and offers advice on troubleshooting upgrade issues.

[Upgrade Guide: XNA Game Studio 3.0 to XNA Game Studio 3.1](#)

Describes how to upgrade your XNA Game Studio 3.0 game to XNA Game Studio 3.1 in Microsoft Visual Studio 2008.

[Troubleshooting Upgrades](#)

Describes common issues with upgrading XNA Game Studio projects.

XACT Update

XNA Game Studio 3.1 includes support for XACT3. XACT 3 has a number of new features, including the following.

Ability to enable a filter on every track.

XACT3 now exposes a filter on every track, letting the sound designer set the filter type and parameters (filter width and the cutoff/center frequency). The filter can be set directly on a track or it can be attached to an RPC. Sound designers either can set specific filter parameters or they can specify a range. When setting a range, they can select a random value for the parameter each time the track is played.

Support for the xWMA compression format.

XACT3 now supports xWMA decoding in software on both Xbox 360 and Windows. xWMA uses the WMA bitstream format in a lightweight wrapper, and it can provide 1.5–2× the compression compared with XMA at similar quality. xWMA is very useful for types of content, such as dialog or music, for which you can afford a small CPU hit to achieve much greater compression. A quality setting similar to XMA's allows you to increase or decrease compression to adjust sound performance.

Changes to the XNA Framework API

Microsoft.Xna.Framework.Content

[ContentSerializerRuntimeTypeAttribute](#)

A custom [Attribute](#) that specifies the corresponding run-time type of this object.

[ContentSerializerTypeVersionAttribute](#)

A custom [Attribute](#) that specifies the corresponding run-time type version of this object.

Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

[ContentTypeWriter.CanDeserializeIntoExistingObject](#)

Determines if deserialization into an existing object is possible.

Microsoft.Xna.Framework.GamerServices

[AvatarAnimation](#)

Provides methods and properties for animating an avatar using standard animations (for example, celebrate).

[AvatarDescription](#)

Provides access to the methods and properties of the description data for an avatar.

[AvatarExpression](#)

Contains the various components of the avatar's face, such as the left and right eyebrows.

[AvatarRenderer](#)

Provides properties and methods for rendering a standard avatar.

[SignedInGamer.AvatarChanged](#)

Occurs when a gamer's avatar changes.

[SignedInGamer.Avatar](#)

Description data for the avatar that represents the gamer.

[SignedInGamer.PartySize](#)

Gets the current party size.

[AvatarAnimationPreset](#)

Defines standard animations for avatars.

[AvatarBone](#)

Defines a list of the useful bones of the avatar model.

[AvatarEyebrow](#)

Defines the standard animation textures for an avatar's eyebrows.

[AvatarEye](#)

Defines the standard animation textures for an avatar's eyes.

[AvatarMouth](#)

Defines the standard animation textures for an avatar's mouth.

[Guide.ShowPartySessions](#)

[Guide.ShowParty](#)

Microsoft.Xna.Framework.Media

[Video](#)

Represents a video.

[VideoPlayer](#)

Provides methods and properties to playback, pause, resume, and stop video. [VideoPlayer](#) also exposes repeat, volume, and play position information.

[VideoSoundtrackType](#)

Type of sounds in a video

Microsoft.Xna.Framework.Net

[LocalNetworkGamer.SendPartyInvites](#)

Sends game invitations to all party members that are not in the current game session.

Microsoft.Xna.Framework.Storage

[StorageDevice.DeleteContainer](#)

Getting Started with XNA Game Studio

XNA Game Studio is an integrated development environment designed to make it easier to develop games for Microsoft Windows, Xbox 360 platforms, and Zune devices. XNA Game Studio extends supported versions of Microsoft Visual Studio tools to support the XNA Framework. The XNA Framework is a managed-code class library that contains functionality targeted specifically to game development tasks. In addition, XNA Game Studio includes tools for incorporating graphical and audio content into your game.

The XNA Framework is designed to be similar to the .NET Framework in terms of its design patterns and idioms. With XNA Game Studio, you are able to incorporate functionality in your game from both the XNA Framework and the .NET Framework. Use the XNA Framework for game-specific tasks such as graphics rendering and managing input, and use the .NET Framework for more general programming tasks.

In This Section

[Documentation Roadmap](#)

Describes the documentation for XNA Game Studio.

[Setup and System Requirements](#)

Describes which operating systems are supported by XNA Game Studio as well as additional software required to use XNA Game Studio.

[Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#)

Describes how to use XNA Game Studio Connect to connect and deploy a game to your Xbox 360 console.

[Connecting to your Zune Device with XNA Game Studio](#)

Describes the steps necessary to connect and deploy a game to your Zune device using XNA Game Studio Connect.

[Your First Game: Microsoft XNA Game Studio in 2D](#)

Describes the steps necessary to create a simple sprite-based game by using XNA Game Studio.

[Going Beyond: XNA Game Studio in 3D](#)

This multipart tutorial takes you through the first steps of creating your own 3D game using XNA Game Studio.

[Upgrading XNA Game Studio Projects](#)

Describes supported and unsupported XNA Game Studio project versions, provides guides for upgrade scenarios, and offers advice on troubleshooting upgrade issues.

[Using XNA Framework Starter Kits](#)

Describes how to create an instance of an XNA Framework Starter Kit.

[Starter Kit: Platformer](#)

Describes the Platformer Starter Kit.

[Support Options and Additional Resources](#)

Describes additional resources for developing with the XNA Framework.

Documentation Roadmap

Describes the documentation for XNA Game Studio.

The XNA Game Studio documentation is divided into the following major sections:

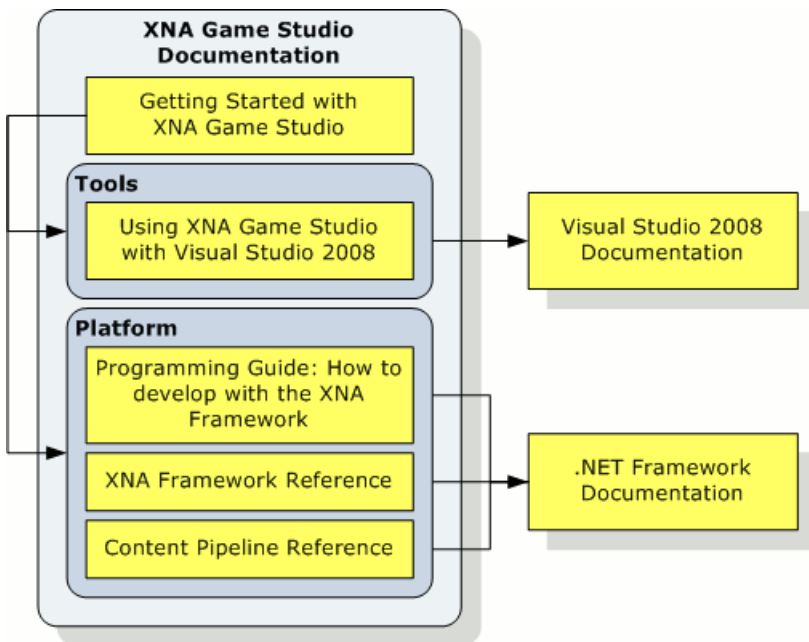
- [Getting Started with XNA Game Studio](#): This section describes how to set up and use XNA Game Studio with Microsoft Visual Studio 2008 and the .NET Framework. It includes information about setting up your Xbox 360 or Zune device to work with Game Studio.

This section links to many topics in the Using XNA Game Studio, Programming Guide, and Reference sections.

- [Using XNA Game Studio](#): This section provides greater detail about working with the Visual Studio Tools and IDE, including how to set up projects for specific platforms, how to work with cross-platform titles, and how to share and distribute your game.
- [Programming Guide](#): This section provides detail about game programming with Game Studio. It describes how to use the XNA Framework classes to implement different features of your game. It is divided into sections based on the feature type, such as graphics, audio, networking, and LIVE programming.
- Reference: The reference section provides even greater detail about the XNA Framework. It is divided into two sub-sections, the [XNA Framework Class Library](#) and [Content Pipeline Class Library](#).

The XNA Game Studio documentation builds upon the existing documentation provided with Microsoft Visual Studio 2008 and the [Microsoft .NET Framework](#). It links to these sets of documentation, where appropriate. You should have access to both of these sets of documentation.

The following diagram shows the relationship between these sections of the documentation, the Visual Studio 2008 documentation, and Microsoft .NET Framework:



Quick Links to XNA Game Studio Documentation

- [Getting Started with XNA Game Studio](#)
- [How to Develop with the XNA Framework](#)
- [XNA Framework Class Reference](#)
- [.NET Compact Framework for Xbox 360](#)
- [Visual C# Developer Center on MSDN](#)
- [.NET Framework Class Reference on MSDN](#)

Setup and System Requirements

Describes which operating systems are supported by XNA Game Studio as well as additional software required to use XNA Game Studio.

In This Section

[Supported Operating Systems and Hardware](#)

Describes the operating systems that fully support XNA Game Studio, and the computing hardware you need to use XNA Game Studio.

[Required Software](#)

Describes the software required to use XNA Game Studio 3.1.

[How To: Configure Help to Use Online Content](#)

Describes how to configure the Visual Studio Help system to use online Help content if such content is available.

[Third-Party Firewall Settings](#)

Describes required settings for third-party firewall products.

Supported Operating Systems and Hardware

Describes the operating systems that fully support XNA Game Studio, and the computing hardware you need to use XNA Game Studio.

Supported Operating Systems

The following operating systems fully support XNA Game Studio.

Operating system	Supported versions
Windows XP	<ul style="list-style-type: none"> • Home Edition • Professional • Media Center Edition • Tablet PC Edition
Windows Vista	<ul style="list-style-type: none"> • Home Basic • Home Premium • Business • Enterprise • Ultimate

Remarks

- XNA Game Studio 3.1 has been tested on Windows 7 Release Candidate. Support for the Windows 7 operating system is expected after the final release.
- Windows Vista Service Pack 1 (or the latest service pack) is supported, but is not required, for XNA Game Studio.
- Windows XP requires, at a minimum, Service Pack 2.
- Although XNA Game Studio itself is available only in English and Japanese, it is supported under any of the available language settings on Windows.
- Administrator permissions are not required to run XNA Game Studio, but they are to install it. For Windows Vista, elevated administrator permissions are required.
- Although the setup program for XNA Game Studio will complete on the Windows Server 2003 and Windows Server 2008 operating systems, XNA Game Studio is not officially supported on these operating systems, and the XNA Game Studio software may not work as expected.

Hardware Requirements

To run XNA Framework games on a computer running a Windows operating system, you need a graphics card that supports, at a minimum, Shader Model 1.1, and DirectX 9.0c. We recommend using a graphics card that supports Shader Model 2.0 because some samples and starter kits may require it.

Ensure your graphics card is using the latest driver. Check with your hardware vendor or access [Microsoft Update](#) to find the latest version of a driver.

Other hardware requirements for XNA Game Studio are identical to those for Visual Studio 2008. For more information, see the [Visual Studio documentation](#).

Required Software

Describes the software required to use XNA Game Studio 3.1.

XNA Game Studio 3.1

To download XNA Game Studio 3.1 itself, go to the [Creators Club Online](#).

Important

You must uninstall previous versions of XNA Game Studio before you install XNA Game Studio 3.1. However, you can install multiple versions of the XNA Framework Redistributable at the same time.

Visual Studio 2008

XNA Game Studio leverages the Visual Studio 2008 development environment, extending it for game development. You can use XNA Game Studio with any supported versions of Visual Studio 2008 tools. Click one of the following links for more information about using XNA Game Studio with Visual Studio 2008.

- [Microsoft Visual C# Express Edition](#)
- [Microsoft Visual Studio 2008](#)

Note

Only members of the Visual Studio 2008 product line support XNA Game Studio 3.1. XNA Game Studio 3.1 is not compatible with Visual Studio 2005 or any previous editions.

XNA Creators Club

To develop games for Xbox 360, you need a membership in the XNA Creators Club. You can get an XNA Creators Club membership from Xbox LIVE Marketplace, or at XNA Creators Club Online: [Sign in](#) and [Membership](#).

XNA Creators Club and Xbox LIVE Membership Requirements

There are two types of online multiplayer games available when you create a networked game with XNA Game Studio: system link game sessions and LIVE sessions. These two forms of online multiplayer gaming describe whether the multiplayer gaming session is on the local area network (system link games), or if the gaming session is advertised and found through the LIVE service. With system link games, you can search the local area network to find the gaming sessions. With LIVE sessions, you can use the LIVE service to find available game sessions. Use the *searchProperties* argument of [NetworkSession.Find](#) to specify the parameters used to search for an available game session—a game session that uses either system or the LIVE service.

Developing and testing a networked game requires at least two machines, but you only need one Xbox 360 console and one Creators Club membership to test network code for the Xbox 360. This is because XNA Framework supports cross-platform system links so developers can run one instance of a game on an Xbox 360, and a second on a Windows-based computer. This functionality helps creators debug their titles without having to purchase a second console. It is also possible for more than one Windows-based development computer to connect several machines in a system link session without any memberships required.

Memberships are required for a player on a Windows-based computer and an Xbox 360 console in a network session, depending on the session type that has been created.

	Xbox 360 console	Windows-based development computer	Zune
Run an XNA Framework Game	LIVE Silver membership + Premium XNA Creators Club membership	No memberships required	No memberships required
Use System Link for Local Area Network gameplay	LIVE Silver membership + Premium XNA Creators Club membership	No memberships required	No memberships required
Sign in to Xbox LIVE and Games for Windows - LIVE Servers	LIVE Silver membership + Premium XNA Creators Club membership	LIVE Silver membership + Premium XNA Creators Club membership	Not available on Zune

Use LIVE to connect to other machines over the Internet while the game is in development	LIVE Gold membership + Premium XNA Creators Club membership	LIVE Silver membership + Premium XNA Creators Club membership	Not available on Zune
⚠Caution			
XNA Framework components that interact with Games for Windows - LIVE require an XNA Creators Club membership. If for any reason a LIVE profile without a Creators Club membership is set to auto sign in to a Windows-based computer, the game will be unable to connect to the LIVE service. Furthermore, the XNA Framework gamer services components, including the Guide, will be unavailable. To disable auto sign in for a LIVE profile on a Windows-based computer, disconnect from the network, launch an XNA Framework game, and then open the Guide.			

Please note the following.

- If you are running an XNA Framework Game on the Microsoft Windows operating system on a dual-core AMD processor, and the game contacts the Xbox LIVE servers or otherwise uses the Gamer Services extensions, you may experience unpredictable behavior such as a hanging of the game process. To work around this issue, you should install the **AMD Dual-Core Optimizer** from AMD on the computer on which you are running the game. Currently, the optimizer is available at the following location.

[AMD Dual-Core Optimizer](#)

Zune Client Version

When developing games for Zune, the Zune client needs to be running at least the 3.0 version of the firmware.

Additional Required Software Installed by XNA Game Studio 3.1 Setup

The XNA Game Studio Setup program also installs the following required software:

DirectX Runtime

XNA Game Studio Setup installs the DirectX runtime, which is required to use the XNA Framework Game API.

Microsoft Cross-Platform Audio Creation Tool

XNA Game Studio Setup installs multiple versions of the Microsoft Cross-Platform Audio Creation Tool (XACT). You can use XACT to author audio content for your game. Games written using the 3.1 XNA Framework are required to use XACT 3 projects, which are not backwards compatible. Games written using the 3.0 XNA Framework are required to use XACT 2 projects. For more information about upgrading your projects in regard to XACT, [Upgrading XNA Game Studio Projects](#).

Games for Windows - LIVE

XNA Game Studio Setup installs Microsoft Games for Windows – LIVE Redistributable, version 2.0.687.0. This software provides support for various gamer services such as retrieving player preferences for local accounts, as well as programmatically displaying various LIVE Guide user interface screens.

Microsoft Visual C# Express Edition

Describes how to use XNA Game Studio with Microsoft Visual C# Express Edition.

Microsoft Visual C# 2008 Express Edition

XNA Game Studio leverages the Visual C# Express development environment, extending it for game development. You can install Visual C# Express from the [Microsoft Express Editions Web site](#).

Microsoft .NET Framework 3.5

XNA Game Studio requires the Microsoft .NET Framework 3.5. The setup program for Microsoft Visual C# 2008 Express Edition will install the .NET Framework if it is not already installed on your computer. You can also install the .NET Framework from the [Microsoft .NET Framework Developer Center](#).

Microsoft Visual Studio 2008

Describes how to use XNA Game Studio with Microsoft Visual Studio 2008.

Microsoft Visual Studio 2008

XNA Game Studio leverages the Microsoft Visual Studio 2008 development environment, extending it for game development. XNA Game Studio works with any of the following Visual Studio 2008 products.

- Visual Studio 2008 Standard Edition
- Visual Studio 2008 Professional Edition
- Visual Studio Team System 2008 Architecture Edition
- Visual Studio Team System 2008 Database Edition
- Visual Studio Team System 2008 Development Edition
- Visual Studio Team System 2008 Test Edition
- Visual Studio Team System 2008 Team Suite

Microsoft .NET Framework 3.5

XNA Game Studio requires that you install the .NET Framework 3.5.

The Setup program for Microsoft Visual Studio 2008 installs the .NET Framework 3.5 if it is not already installed on your computer. You can also install the .NET Framework from the [Microsoft .NET Framework Developer Center](#).

How To: Configure Help to Use Online Content

Describes how to configure the Visual Studio Help system to use online Help content if such content is available.

Configuring Visual Studio Help

To configure Visual Studio Help to use online content

1. On the **Tools** menu, click **Options**.

The **Options** dialog box appears.

2. In the tree view control in the left pane of the **Options** dialog box, click the **Help** node beneath the **Environment** node.

You may need to check the **Show all settings** checkbox to make the **Help** node visible.

3. Click the **Online** node beneath the **Help** node.

You may need to click the plus sign next to the **Help** node to make the **Online** node visible.

4. In the set of options labeled **When loading Help content**, select **Try online first, then local**.

5. Click **OK**.

Third-Party Firewall Settings

The XNA Game Studio setup program adjusts Windows Firewall settings to allow the XnaTransX.exe application to send and receive network traffic. This is necessary for your Windows-based development computer to communicate with the Xbox 360 console. Similarly, the setup program adjusts Windows Firewall settings for the XnaLiveProxy.exe application to enable network communications for networked games.

If you are using a firewall product other than Windows Firewall—that is, a third-party firewall setting—you will need to configure this firewall to allow XnaTransX.exe and XnaLiveProxy.exe to send and receive network traffic. Some firewall products enable you to unblock entire applications, some enable you to unblock specific ports, and some, such as Windows Firewall, enable you to do both.

To unblock the entire XnaTransX.exe application, specify the following executable:

```
<install dir>\Common Files\Microsoft Shared\XNA\XnaTrans\v3.1\XnaTransX.exe
```

For the XnaLiveProxy.exe application, specify the following executable:

```
<install dir>\Microsoft XNA\XNA Game Studio\v3.1\Bin\XnaLiveProxy.exe
```

Note

In the above paths, <install dir> refers to the directory beneath which XNA Game Studio is installed. By default, this directory is %ProgramFiles%.

To unblock specific ports, configure your firewall using the following information.

Application	Port	Protocol	Inbound or Outbound
XnaTransX.exe	1001	TCP	outbound
XnaTransX.exe	1001	UDP	inbound/outbound
XnaTransX.exe	3825	UDP	inbound/outbound
XnaTransX.exe	3835	UDP	inbound/outbound
XnaLiveProxy.exe	1000	UDP	inbound/outbound
XnaLiveProxy.exe	3074	UDP	inbound/outbound

Firewall Settings for Connectivity to Xbox LIVE

You may also need to unblock ports on firewall devices between your Xbox 360 console and the external network to allow the console to connect to the Xbox LIVE service. For more information, see the following Knowledge Base article: [Xbox 360: Port settings for Xbox LIVE](#)

Connecting to Your Xbox 360 Console with XNA Game Studio 3.1

Describes how to use XNA Game Studio Connect to connect and deploy a game to your Xbox 360 console.

XNA Game Studio makes it easy for you to create games for both your Windows-based computer and your Xbox 360 console. By joining the XNA Creators Club as a premium member and downloading XNA Game Studio, you can begin developing for your Windows-based computer. To begin developing for Xbox 360, follow these additional steps.

This guide assumes you have installed a supported version of Visual Studio tools and XNA Game Studio already. You must install these products before you connect to an Xbox 360 console. For information about installing these products, see [Setup and System Requirements](#).

- [Step 1: Sign In to Xbox LIVE](#)
- [Step 2: Download XNA Game Studio Connect](#)
- [Step 3: Connect Your Xbox 360 Console and Windows-Based Computer](#)
- [Step 4: Create and Deploy an Xbox 360 Project](#)
- [What's Next?](#)

Step 1: Sign In to Xbox LIVE

Turn on your Xbox 360 console, and sign in to Xbox LIVE. At the very least, you will need a Silver Xbox LIVE membership, an XNA Creators Club premium membership, and a hard drive for your Xbox 360 console to be able to develop games for Xbox 360 using XNA Game Studio. While you are in XNA Game Studio Connect or playing an XNA Game Studio game, you need to be connected to Xbox LIVE.

Connections between your Xbox 360 console and XNA Game Studio require a premium membership in XNA Creators Club. To sign up, visit the [XNA Creators Club Web site](#).

Step 2: Download XNA Game Studio Connect

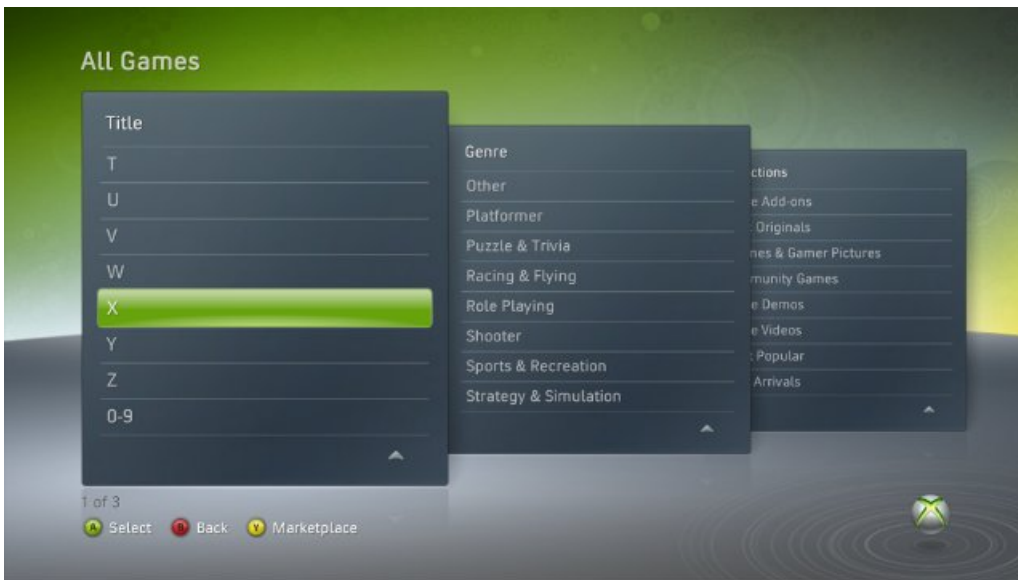
You must download XNA Game Studio Connect from Xbox LIVE Marketplace, and install it on the Xbox 360 console. Go to the Xbox LIVE Marketplace to find XNA Game Studio Connect. To download XNA Game Studio Connect, choose one of the following options:

Browsing to All Games using the Guide

- From the Guide, navigate to the **Marketplace** blade.
- Select **Game Marketplace**, then **All Games**. This brings you to the **All Games** screen.
- Continue with *Downloading XNA Game Studio Connect from the All Games screen*, below.

Browsing to All Games using the New Xbox Experience (NXE) UI

- Navigate to **Game Marketplace**, and select **Explore Game Content**. Press **A** on the controller.
- Select **Browse**, and press **A**. This brings you to the **All Games** screen.
- Continue with *Downloading XNA Game Studio Connect from the All Games screen*, below.



Once you've arrived at the **All Games** screen from either the Guide or NXE, use the following procedure to download XNA Game Studio Connect.

Downloading XNA Game Studio Connect from the All Games screen

- From **All Games**, browse to the **Genre** screen, and select **Other**.

Note

You can also do this by scrolling down the **Title** list, and selecting **X**.

- Scroll to **XNA Creators Club**, and press **A**.
- From the **XNA Creators Club** pane, select **All Downloads**, then **XNA Game Studio Connect**.
- Press the **A** controller button, and select **Confirm Download** to begin downloading.

Step 3: Connect Your Xbox 360 Console and Windows-Based Computer

When you develop games for Xbox 360, you develop them on your Windows-based computer, then transfer them over your local network to your Xbox 360 console. This requires that your Windows-based computer and Xbox 360 console share the same subnet. Most home networking layouts support this configuration. If your console and computer share a router or hub, it is likely that they share the same subnet.

With your computer and console on the same subnet, follow these steps to set up a connection between your computer and your console.

Step 3A. On Your Xbox 360 Console, Generate a Connection Key:

1. From the Xbox Dashboard, go to **My Xbox**, select **Game Library**, and press the **A** controller button.

Also, you have the option of using the Xbox Guide: select the **Games** tab, then **Game Library**, and press the **A** controller button.

2. From the **Game Library**, go to the **Collections** tab, select **Community Games**, and press **A**.
3. Select **XNA Game Studio Connect**, and press **A**.
4. Select **Launch**, and press **A**.

The XNA Game Studio Connect screen appears.



5. If the XNA Game Studio Connect screen displays a connection key, continue to step 3B.

If the connection key does not appear, you can generate a new key by pressing the **X** controller button.

If the connection key does not appear, the Xbox console could already be connected to this Windows-based computer. XNA Game Studio allows multiple connection keys for multiple users on multiple computers. For more information, see [Using XNA Game Studio Device Center](#). To add a new connection key to the list of connection keys recognized by this Xbox 360 console, press **X**. To reset all connection keys and generate a new connection key to connect to this Xbox 360 console, press **Y**.

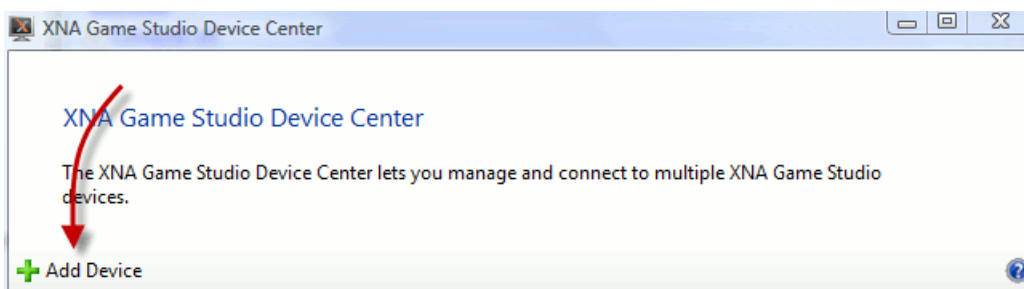
Step 3B. On Your Windows-Based Computer, Enter the Connection Key and Initiate the Connection

1. From the **Start** menu, select **Programs**, select **XNA Game Studio 3.1**, and launch the **XNA Game Studio Device Center**.

Note

You can also use the **XNA Game Studio Device Management** toolbar to directly launch XNA Game Studio Device Center and add a device. Visual Studio 2008 displays the **XNA Game Studio Device Management** toolbar by default when an Xbox 360 game is open in Visual Studio.

2. Click **Add Device**.



3. Select the type of device you're adding. In this case, click **Xbox 360**.



4. Enter a name for this Xbox 360 console, and click **Next**.

Give your Xbox 360 console a name.

Please choose a name for your Xbox 360 console. This will be used to identify the Xbox 360 console in the XNA Game Studio Device Center.

Xbox 360 Name:

This name serves only to identify your Xbox 360 console to XNA Game Studio. The name does not need to correspond to any other computer or Xbox 360 name.

5. Enter the connection key that is displayed in XNA Game Studio Connect on the Xbox 360.

Type your Connection Key

You can find the Connection Key displayed on the XNA Game Studio Connect screen on your Xbox 360 console.

The Connection Key looks similar to this:

XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Type your connection key (dashes will be added automatically):

Note

The connection key might be somewhat hard to read on a standard television screen. The following guide should help you identify specific letters and numbers:

- The number "1" has a small tick at its top left; the capital letter "l" does not.
- The capital letter "B" has a straight line on the left; the number "8" does not.
- The number "3," sometimes mistaken for a "B," also has no straight side on the left.
- The number "0" and the capital letter "O" are so similar that XNA Game Studio Connect treats these characters as the same. Therefore, the number "0" and the capital letter "O" are interchangeable.

If the connection key is still too difficult to read, press **X** on the Xbox 360 controller to generate a new connection key.

6. Once you are sure that the two keys match, click **Next** on the **XNA Game Studio Devices** dialog box.

XNA Game Studio Device Center will test the connection with the Xbox 360 console.

If the connection is successful, the XNA Game Studio Device Center on the Windows-based computer will display "Successfully connected to the Xbox 360 console." XNA Game Studio Connect on the Xbox 360 console will display "Waiting for computer connection," followed by the name you have chosen for your Xbox 360 console in the XNA Game Studio Device Center.

If the XNA Game Studio Device Center fails to connect to the Xbox 360 console, click **Try again** to edit the connection key and try again. If the connection continues to fail, make a careful note of the error message displayed at the bottom of the **XNA Game Studio Devices** dialog box. This error message can help you or a technician diagnose the cause of the connection failure, if it did not result from mismatched keys. For more information about troubleshooting a failed connection, see [Troubleshooting Xbox 360 Game Deployment](#).

7. Click **Finish**.

The name you gave to your Xbox 360 console will be listed in the XNA Game Studio Device Center. From now on, your computer and your console can connect to each other easily.

Step 4: Create and Deploy an Xbox 360 Project

Try out the Xbox 360 console by deploying a simple, blank XNA Game Studio game to it.

1. On your Windows-based computer, at the main Visual Studio screen, select the **File** menu, and then click **New Project**.

- From **Project types**, expand the **Visual C#**, and click **XNA Game Studio 3.1**.
- In **Templates**, select **Xbox 360 Game (3.1)**, and then click **OK**.

At this point, you should bring up XNA Game Studio Connect to prepare the Xbox 360 console to receive content from the Windows-based computer.

- From the Xbox Dashboard, go to **My Xbox**, select **Game Library**, and press **A** on the controller.
You can also use the Guide: select the **Games** tab, select **Game Library**, and then press **A**.
- From the **Game Library**, go to the **Collections** tab, select **Community Games**, and press **A**.
- Select **XNA Game Studio Connect**, and press **A**.
- Select **Launch**, and press **A**.

The XNA Game Studio Connect screen appears.



- On your Windows-based computer, with your new project open, press the F5 key.

The project will build, deploy necessary files to the Xbox 360 console, and run.

At this point, you should see a simple display on your console—just a blue screen. If you see this screen, you have successfully deployed a game to your Xbox 360 console. You can now deploy and play any game you create on your console.

- To stop the game and return to the main screen of XNA Game Studio Connect, either press the **BACK** button on your Xbox 360 gamepad, or press SHIFT+F5 to stop debugging on your computer.

If the deployment fails, you may need to check your network settings. See [Troubleshooting Xbox 360 Game Deployment](#) for more information.

What's Next?

You have successfully deployed your first Xbox 360 game. This game, like every game you deploy, will now appear in the **Recent Games** tab in the **Game Library** on your Xbox 360 console. It can also be found by selecting **Community Games** in the **Collections** tab in your **Game Library**. Using **Game Library**, you can play any of the games you have deployed to your Xbox 360 console. From **Game Library**, you can also delete games you do not want to keep.

Once your game is loaded onto the hard disk so that it appears in the **Recent Games** tab of the **Game Library** on your Xbox 360 console, it can be transferred to a memory unit if one is installed. From **System Settings** on your Xbox 360 console, the **Memory** selection enables you to transfer a game from the hard disk to the memory unit. The memory unit can then be installed on any other Xbox 360 console and played or copied.

Important

Only XNA Framework games built and deployed with XNA Game Studio 3.0 and later can be transferred to a memory unit. If you wish to transfer a game developed under an earlier version of XNA Game Studio, you must convert, build, and deploy the project using the current version of XNA Game Studio.

A good next step would be to try some of the tutorials. The code used in the tutorials works on both Windows-based computers and on the Xbox 360 console, so you can try either platform to get a feel for how to develop for both. Start with [Tutorial 1: Displaying a 3D Model on the Screen](#) for your first taste of game development.

See Also [Troubleshooting Xbox 360 Game Deployment](#)

[Deploying an Xbox 360 Game](#)

[Using XNA Game Studio Device Center](#)

[Third-Party Firewall Settings](#)

Connecting to your Zune Device with XNA Game Studio

Describes the steps necessary to connect and deploy a game to your Zune device using XNA Game Studio Connect.

XNA Game Studio allows you to easily create games for your Zune device, Windows-based computer, and Xbox 360 console. Follow this guide to develop games for the Zune platform topic.

Note

This guide assumes you have already installed a supported version of Visual Studio and XNA Game Studio. You must install these products before you connect to a Zune device. For information about installing these products, see [Setup and System Requirements](#).

- [Step 1: Update Your Zune Device Firmware](#)
- [Step 2: Add Your Zune in the XNA Game Studio Device Center](#)
- [Step 3: Create and Deploy a Zune Project](#)
- [What's Next?](#)

Step 1: Update Your Zune Device Firmware

XNA Game Studio is supported on Zune devices running firmware version 3.0 or later. If your device does not already have firmware version 3.0 or later, you'll need to update your Zune firmware to the most current version.

To determine Zune device firmware version

- On the Zune device, select **settings | about | zune**.

The firmware version will be displayed.

If you have a version older than 3.0, you'll need to update your Zune.

To update your Zune device

1. Make sure you have the current version of the Zune software installed on your desktop computer.
 - If you are a new Zune customer, visit <http://www.zune.net/setup> and install the Zune software for your computer.
 - If you already have the Zune software installed on your computer, check for updates. From the Zune software, click **Settings | GENERAL | CHECK FOR UPDATES**.
2. Use the Zune sync cable to connect the Zune device to your computer. Connect the sync cable to a USB port on the computer, not a USB hub.

Note

If the Zune device battery is critically low, we recommend that you charge your Zune device before you continue. For more information about how to charge your Zune device, see the following Microsoft Knowledge Base article: [How to charge the battery in your Zune device](#).

3. Open the Zune software on your desktop computer while the Zune device is still connected to your computer.

The Zune software searches the Internet for the latest version of the firmware. If a newer version of the Zune device firmware is available, you will get a notice to update your device.

Note

Do not move or disconnect the Zune device until the update is complete. The update may take several minutes. During this time, the Zune device may restart several times. Please be patient.

You can also manually check for new device updates, if desired. To do so: from the Zune software on your desktop computer, click **Settings | device | DEVICE UPDATE**.

For more information about updating your Zune device, or to troubleshoot issues with updating, please see the following Microsoft Knowledge Base article: [How to update your Zune device](#).

After you update your Zune firmware to version 3.0 or later, you won't need to perform this step again.

Step 2: Add Your Zune in the XNA Game Studio Device Center

Once you add your Zune to the XNA Game Studio Device Center, XNA Game Studio will be able to connect to your Zune so that you can deploy and debug games.

To add your Zune to the XNA Game Studio Device Center

1. Use the Zune sync cable to connect the Zune device to your computer.
2. Start the device center by clicking **Start, All Programs, Microsoft XNA Game Studio 3.1**, and then click **XNA Game Studio Device Center**.
3. Click **Add Device**, and then click **Zune**.

The device center will display a list of connected Zune devices.

4. In the list of connected Zunes, click the one that you want to add.
5. Click **Next**.

The Zune you selected will be added to the list of devices in the device center.

6. Close the XNA Game Studio Device Center.

Note

If you are having trouble connecting to your Zune device, or the device is not showing up in the device center, close the Zune client desktop software (if it is running).

You can also configure the Zune desktop software so it won't launch automatically when a Zune device is connected. This setting is located under **settings | device | SYNC OPTIONS**.

Step 3: Create and Deploy a Zune Project

Now you can try out Zune development by deploying a simple, blank XNA Game Studio game.

To deploy a simple, blank XNA Game Studio game

1. On your Windows-based computer, at the main Visual Studio screen, click **File | New Project** (or **File | New | Project**, depending on your Visual Studio configuration).
2. From **Project types**, expand **Visual C#** and click **XNA Game Studio 3.1**.
3. In **Visual Studio installed templates**, click **Zune Game (3.1)**, and then click **OK**.
4. On your Windows-based computer, with your new project open, press the F5 key.

The project will build. It launches XNA Game Studio Connect on the Zune device, deploys the project and other necessary files to the Zune, and then launches the game.

At this point, you should see a simple display on your Zune: a light blue screen. If you see this screen, you have successfully deployed a game to your Zune. You can now deploy and play any game you dream and build.

5. To stop the game and return to the main screen of XNA Game Studio Connect, either:
 - Press the **BACK** button on your Zune.
 - or-
 - Stop debugging by pressing SHIFT+F5 on your computer.

Note

When an XNA Framework game exits on the Zune, the Zune device reboots. This behavior is by design—it ensures that game and run-time resources are completely cleaned up after a game exits.

Note

Holding down the Back button for 2 seconds forces the currently executing game to exit, regardless of the game's normal exit semantics.

What's Next?

You have successfully deployed your first Zune game. This game, like the other games you deploy to your Zune, will appear under **games** on your Zune device. Using the **games** menu, you can play or delete the games you have deployed to your Zune.

A good next step would be to try the 2D tutorial [Your First Game: Microsoft XNA Game Studio in 2D](#). The code in the 2D tutorial works on Zune, Windows-based computers, and on the Xbox 360 console, so you can try any or all of the platforms to get a feel for developing on each.

For an overview of Zune-specific features and programming concepts, see [Zune Programming Considerations](#).

Note

See the *Supported Platforms* section on individual reference pages to check for Zune support. If Zune is not on the supported platforms list, the API is not supported. If listed, there is support for the API.

See Also [Zune Programming Considerations](#)
[Your First Game: Microsoft XNA Game Studio in 2D Programming Guide](#)

Your First Game: Microsoft XNA Game Studio in 2D

Describes the steps necessary to create a simple sprite-based game by using XNA Game Studio.

Note

A sprite is a simple 2D graphic (such as a bitmap) that is displayed on the screen using a call to [SpriteBatch.Draw](#).

- [The Complete Sample](#)
- [Step 1: Install Your Software](#)
- [Step 2: Create a New Project](#)
- [Step 3: View the Code](#)
- [Step 4: Add a Sprite](#)
- [Step 5: Make the Sprite Move and Bounce](#)
- [Step 6: Explore!](#)

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download MyFirstGame_Tutorial_Sample.zip](#).

Step 1: Install Your Software

Before you begin, make sure that you have installed all the necessary software, including a supported version of Microsoft Visual Studio tools and XNA Game Studio. See [Required Software](#) for a list of required programs.

Step 2: Create a New Project

1. From the **Start** menu, click **All Programs**, click the **XNA Game Studio 3.1** folder, and then click your supported version of Microsoft Visual Studio tools.
2. When the Start Page appears, click the **File** menu, and then click **New Project**.
A dialog box appears with a tree list on the left pane, marked Project Types.
3. Select the **XNA Game Studio 3.1** tree node underneath the **Visual C#** node.
A set of available projects appears in the right pane.
4. In the right pane of the dialog box that appears, click **Windows Game (3.1)**, and then type a title for your project (such as "MyFirstGame") in the **Name** box.
5. Type a path where you'd like to save your project in the **Location** box, and then click **OK**.

After creating a new project, you'll be presented with the code view of your game.

Step 3: View the Code

Some of the hard work has already been done for you. If you build and run your game now, the [GraphicsDeviceManager](#) will set up your screen size and render a blank screen. Your game will run and update all by itself. It's up to you to insert your own code to make the game more interesting.

Much of the code to start and run your game has already been written for you. You can insert your own code now.

- The [Initialize](#) method is where you can initialize any assets that do not require a [GraphicsDevice](#) to be initialized.
- The [LoadContent](#) method is where you load any necessary game assets such as models and textures.
- The [UnloadContent](#) method is where any game assets can be released. Generally, no extra code is required here, as assets will be released automatically when they are no longer needed.
- The [Update](#) loop is the best place to update your game logic: move objects around, take player input, decide the outcome of collisions between objects, and so on.
- The [Draw](#) loop is the best place to render all of your objects and backgrounds on the screen.

Step 4: Add a Sprite

The next step is to add a graphic that can be drawn on the screen. Use a small graphics file, such as a small .bmp or .jpg file. Be creative—you can even make your own. You can even skip ahead a bit and make a sprite that "hides" parts that should not be seen (such as edges or corners) so that it looks even better.

Once you have a graphic picked out on your computer, follow these steps.

1. Make sure you can see the Solution Explorer for your project on the right side of the window. If you cannot see it, click the **View** menu, and then click **Solution Explorer**.

When it appears, you will see files associated with your project in a tree structure. Inside the tree, you will see a node named **Content**.

2. Right-click the **Content** node, click **Add**, click **Existing Item**, and then browse to your graphic.

If you can't see any files, make sure you change the **Files of type** selection box to read **Texture Files**.

3. Click the graphic file, and then click **Add**.

An entry for the graphic file will appear in Solution Explorer.

4. Click the entry for the graphic in the Solution Explorer. If you do not see the entry, ensure the **Content** node is expanded by clicking the small plus sign (+) to the left of the node, then click on the entry that appears underneath the **Content** node.

When you add a graphic file, it is automatically added to the XNA Framework Content Pipeline, which will allow you to quickly and easily load the graphic into your game.

In the **Properties** window below Solution Explorer, look for the "Asset Name" property. Note the name; you'll use it in your code to load the graphic so it can be displayed in your game.

5. If the **Properties** window is not visible, press F4, or click the **View** menu, and then click **Properties Window**.

Now, you must write code that loads and displays the sprite on the screen. You can use the instructions in [How To: Draw a Sprite](#), or follow along here.

6. Back in the Code view of your game, find the `LoadContent` method, and add the following lines in and above the method so it looks similar to this:

C#

```
// This is a texture we can render.
Texture2D myTexture;

// Set the coordinates to draw the sprite at.
Vector2 spritePosition = Vector2.Zero;

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    myTexture = Content.Load<Texture2D>("mytexture");
}
```

Make sure the call to `ContentManager.Load` is using the "Asset Name" you saw in the Properties window in the previous step. This code will load and prepare your graphic to be drawn, and will reload your graphic if the graphics device is reset (such as in the case of the game window being resized).

7. Now, add code to the `Draw` loop so it looks like this:

C#

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    // Draw the sprite.
```

```

        spriteBatch.Begin(SpriteBlendMode.AlphaBlend);
        spriteBatch.Draw(myTexture, spritePosition, Color.White);
        spriteBatch.End();

        base.Draw(gameTime);
    }

```

This code draws the sprite on the screen each frame.

Notice the parameter passed by the `Begin` method, `SpriteBlendMode.AlphaBlend`. This parameter tells the `Draw` method to use the alpha channel of the source color to create a transparency effect so that the destination color appears through the source color. For more information, see [What is Color Blending](#).

8. Build and run your game.

The sprite appears.

Now, it's time to give it some motion.

Step 5: Make the Sprite Move and Bounce

- Change the lines of code in and below the `Update` method to read this way:

C#

```

// Store some information about the sprite's motion.
Vector2 spriteSpeed = new Vector2(50.0f, 50.0f);

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    // Move the sprite around.
    UpdateSprite(gameTime);

    base.Update(gameTime);
}

void UpdateSprite(GameTime gameTime)
{
    // Move the sprite by speed, scaled by elapsed time.
    spritePosition +=
        spriteSpeed * (float)gameTime.ElapsedGameTime.TotalSeconds;

    int MaxX =
        graphics.GraphicsDevice.Viewport.Width - myTexture.Width;
    int MinX = 0;
    int MaxY =
        graphics.GraphicsDevice.Viewport.Height - myTexture.Height;
    int MinY = 0;

    // Check for bounce.
    if (spritePosition.X > MaxX)
    {
        spriteSpeed.X *= -1;
        spritePosition.X = MaxX;
    }

    else if (spritePosition.X < MinX)

```

```
{
    spriteSpeed.X *= -1;
    spritePosition.X = MinX;
}

if (spritePosition.Y > MaxY)
{
    spriteSpeed.Y *= -1;
    spritePosition.Y = MaxY;
}

else if (spritePosition.Y < MinY)
{
    spriteSpeed.Y *= -1;
    spritePosition.Y = MinY;
}
}
```

This adds a little bit of logic that will move the sprite around each frame and cause the sprite to change direction if it hits the edges of the game window.

- Build and run your game.

The sprite moves across the screen and changes direction when it encounters the edges of the game window.

Step 6: Explore!

From here, you can do just about anything. If you're ready to jump into a more complex example, including 3D graphics, input, and audio, see [Going Beyond: XNA Game Studio in 3D](#).

Here are some more ideas to extend this sample:

- Experiment with the call to [SpriteBatch.Draw](#). Change the *Color* parameter to tint the sprite. See [How To: Tint a Sprite](#).
- Add a second sprite, and use [BoundingBox](#) objects to allow the sprites to collide with one another. (See [How To: Detect Whether Two Models Collide](#).)
- Use [Keyboard](#), [Mouse](#), or [GamePad](#) to make the sprite respond to movements of an input device. (See [Input Overview](#).)
- Create some audio events so that the sprite makes sounds as it moves. (See [How To: Add a Sound File to Your Game Using XACT](#) and [How To: Play a Sound Using XACT](#).)
- Instead of a sprite, use a 3D primitive that moves around in 3D space. (See [How To: Draw Points, Lines, and Other 3D Primitives](#).)
- Get more ideas and resources at [XNA Creators Club Online](#).

Going Beyond: XNA Game Studio in 3D

This multipart tutorial takes you through the first steps of creating your own 3D game using XNA Game Studio.

In XNA Game Studio, it is easier than ever to create games for Windows and for Xbox 360. This tutorial highlights how to load 2D and 3D resources into your game using the XNA Framework Content Pipeline; how to take user input from an Xbox 360 Controller; and how to use the Microsoft Cross-Platform Audio Creation Tool (XACT) to create audio for your game. Each step of the way, you'll have something to see or hear so you will know you are making progress.

Note

The code displayed in each section will work on both Xbox 360 and Windows, but not on Zune. Zune devices do not support 3D graphics, and XACT is not available on Zune. To set up your Xbox 360 console to work with XNA Game Studio, see [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#).

In This Section

[Tutorial 1: Displaying a 3D Model on the Screen](#)

This article details how to use the XNA Framework Content Pipeline to load a 3D model and its associated textures, and it presents the code necessary to display the model on the screen.

[Tutorial 2: Making Your Model Move Using Input](#)

This article details how to use the XNA Framework Input API to take user input from an Xbox 360 Controller and apply it to the model displayed in Tutorial 1.

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Details how to use the XNA Framework Audio API to play them.

[Tutorial 4: Make a Game in 60 Minutes](#)

This tutorial helps you learn about the process of game construction, while guiding you through writing a relatively complete game.

[Tutorial 5: Adding Multiplayer and Networking Support to the Game](#)

This tutorial adds two-player competitive game play to the game completed in Tutorial 4.

See Also

Tasks

[FuelCell](#)

[Getting Started with 2D Games at XNA Creators Club Online](#)

[Getting Started with 3D Games at XNA Creators Club Online](#)

Tutorial 1: Displaying a 3D Model on the Screen

This article details how to use the XNA Framework Content Pipeline to load a 3D model and its associated textures, and it presents the code necessary to display the model on the screen.

- [The Complete Sample](#)
- [Introduction](#)
- [Step 1: Downloading the Art Assets](#)
- [Step 2: Creating the New Project](#)
- [Step 3: Load the Model by Using the Content Pipeline](#)
- [Step 4: Display the Model on the Screen \(and Make It Rotate\)](#)
- [Congratulations!](#)
- [Ideas to Expand](#)

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download GoingBeyond1_Tutorial_Sample.zip.](#)

Introduction

In [Your First Game: Microsoft XNA Game Studio in 2D](#), you saw a simple example that used the XNA Framework Content Pipeline to load a sprite, represented by a [Texture2D](#) object. You also used the XNA Framework to draw the sprite on the screen. This tutorial goes beyond that simple sample to help introduce you to many concepts that XNA Game Studio makes easy, so you can focus on making fun, interactive games.

This first tutorial will introduce you to the Content Pipeline in a little more detail, and will introduce you to some of the XNA Framework API calls you will use to draw 3D objects on the screen. When you complete this tutorial, you'll have a 3D model drawing on your screen. The model will have textures and lighting. Let's get started!

Step 1: Downloading the Art Assets

The first thing that you will need before you start coding are some art assets to play around with. In this case, you need a 3D model, and an associated texture file so that the model has some detail. These assets will be loaded into your game using the XNA Framework Content Pipeline, which is a feature built right into the Solution Explorer feature of supported version of Microsoft Visual Studio tools.

You can find these assets in this sample file ([GoingBeyond1_Tutorial_Sample.zip](#)). Download the sample file now and extract its contents to a directory on your local drive.

Step 2: Creating the New Project

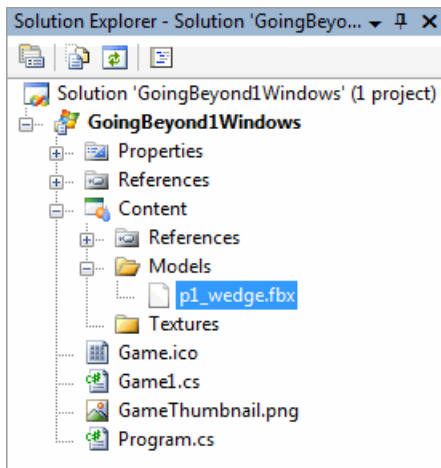
Now that the art assets are available to you, the next step is to create the actual code project that you will be writing.

- Click the **File** menu, and then click **New Project** to create a new project. A dialog box will appear with a tree list on the left pane, marked **Project Types**.
- Select the **XNA Game Studio 3.1** tree node underneath the **Visual C#** node. A set of available projects will appear in the right pane.
- In the right pane of the dialog box, click either **Windows Game (3.1)** or **Xbox 360 Game (3.1)**, depending on whether you are developing on the Xbox 360 or Windows. If you develop for Xbox 360, be sure you have a membership in the XNA Creators Club as described in [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#) Otherwise, you will not be able to play your game!
- Type a name for your game into the **Name** field, and a path to where you want the game files stored in the **Location** field.
- Click **OK**.

The code for your new game will be displayed. The project already contains many of the methods that are needed to start and run a game. Right now, however, you need to make sure your art assets are being loaded. Then you can modify the game to display them on the screen. Follow these steps to get some art into your project.

- Make sure you can see the **Solution Explorer** for your project on the right side of the window. If you cannot see it, click the **View** menu, and then click **Solution Explorer**. When it appears, you will see files associated with your project in a tree structure.
- In **Solution Explorer**, look for a node named **Content**. This is where you will store the art and audio for your game. You must add two folders underneath this one.
- Right-click the **Content** node, click **Add**, and then click **New Folder**. This will create a new folder under the **Content** node. Name this folder Models.
- Repeat the last step, creating a new folder under the **Content** node. This time, call the folder Textures.

Your project structure should look similar to this:



You are now ready to add the art from the previously downloaded sample file. The first is a 3D model that will go into this new Content\Models folder, and the second is a texture that will be drawn on the 3D model; this will go in the Content\Textures folder. The files you need are included in the Content sub-project from the GoingBeyond1_Tutorial_Sample.zip file. To add them:

- Right-click the Models folder in the **Solution Explorer**, click **Add**, and then click **Existing Item**. Using the dialog box that appears, browse back to the path you extracted the contents of the sample file to and find the Contents\Models folder. Select **p1_wedge.fbx**. If you cannot see any files, make sure you change the **Files of type** selection box to read **Model Files**. Click **OK**.
- Now, copy the texture associated with the model into the Textures folder. To do this, open a Windows Explorer window and browse to the Content\Textures folder of the extracted sample. Copy wedge_p1_diff_v1.tga, and then browse to your project folder, then into the Content\Textures folder, and paste in the file you just copied.

Note that you do not see the texture you added in **Solution Explorer**. When you add a model, the textures that the model uses do not need to be added to the Content Pipeline. If you need to add textures that you will access manually (such as textures used for 2D sprite drawing), do so via **Solution Explorer**. Otherwise, you can simply copy the texture files to the appropriate folder.

When the files are added to the project, the Content Pipeline automatically identifies them as content files and sets the appropriate processors to run when you build your project. This will happen silently; you will not need to do anything. If you would like to learn more about the Content Pipeline, see [Content Pipeline](#).

Note

Model files contain path information for the textures they use. The ship model you recently added to your project expects to find its texture in a Textures folder that exists alongside the folder the model is in. This will be true for all models used in this tutorial. Models you create or retrieve from other sources may have different path requirements and therefore may require different folder setups. Determine the correct texture paths by examining the model files, or compiling the model as part of your game using XNA Game Studio and noting any Content Pipeline path errors that are returned.

At this point, you are ready to code.

Step 3: Load the Model by Using the Content Pipeline

Take a look at the code for Game1.cs. It should still be on your screen from opening up your project. You will see a lot already done for you. Each of the methods already in the code are waiting for you to drop in your own calls to the XNA Framework. For now, start by modifying the [LoadContent](#) method.

- In the code, find the [LoadContent](#) method.
- Modify the code (including adding the lines shown above the method) to look like this:

C#

```
// Set the 3D model to draw.
Model myModel;

// The aspect ratio determines how to scale 3d to 2d projection.
float aspectRatio;

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    myModel = Content.Load<Model>("Models\\p1_wedge");
    aspectRatio = graphics.GraphicsDevice.Viewport.AspectRatio;
}
}
```

In that step, you have told the Content Pipeline to load your model into your game when [LoadContent](#) is called at the beginning of your game. Note how you have to pass in the path to the asset relative to your project directory. Also note that there is no extension on the asset anymore. The name of the asset can be anything you want, but by default it is the name of the asset file minus its extension. To see more information on how to change the name of your asset, see [Game Asset Properties](#).

The code now loads the model. Your next step is to get it showing on the screen.

Step 4: Display the Model on the Screen (and Make It Rotate)

You will want to modify two of the methods in your Game1.cs file.

- In the [Draw](#) method, you will draw the model on the screen with texture and lighting.
- In the [Update](#) method, you will make the model change its orientation based on time, so it appears to rotate over time.

Do the harder work first—drawing the model. The first step is to use some XNA Framework methods to set up the model's position and lighting.

- In the code, find the [Draw](#) method.
- Modify the code (including adding the lines shown above the method) to look like this:

C#

```
// Set the position of the model in world space, and set the rotation.
Vector3 modelPosition = Vector3.Zero;
float modelRotation = 0.0f;

// Set the position of the camera in world space, for our view matrix.
Vector3 cameraPosition = new Vector3(0.0f, 50.0f, 5000.0f);

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    // Copy any parent transforms.
    Matrix[] transforms = new Matrix[myModel.Bones.Count];
    myModel.CopyAbsoluteBoneTransformsTo(transforms);

    // Draw the model. A model can have multiple meshes, so loop.
    foreach (ModelMesh mesh in myModel.Meshes)
    {

```

```

// This is where the mesh orientation is set, as well
// as our camera and projection.
foreach (BasicEffect effect in mesh.Effects)
{
    effect.EnableDefaultLighting();
    effect.World = transforms[mesh.ParentBone.Index] *
        Matrix.CreateRotationY(modelRotation)
        * Matrix.CreateTranslation(modelPosition);
    effect.View = Matrix.CreateLookAt(cameraPosition,
        Vector3.Zero, Vector3.Up);
    effect.Projection = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.ToRadians(45.0f), aspectRatio,
        1.0f, 10000.0f);
}
// Draw the mesh, using the effects set above.
mesh.Draw();
}
base.Draw(gameTime);
}

```

This code uses helper methods provided by the XNA Framework to set up the necessary 3D math and lighting to display the model on the screen. Use the [World](#) matrix to change the position of the model in the world, the [View](#) matrix to change the position and direction of the camera (your eye), and the [Projection](#) matrix to control how the view of the 3D world is turned into a 2D image (projected) on your screen.

The call to [CopyAbsoluteBoneTransformsTo](#) and associated code inside the setup of the [World](#) matrix are not strictly necessary for this model. However, when using more complicated models, which often use hierarchical structure (where mesh positions, scales, and rotations are controlled by "bones"), this code ensures that any mesh is first transformed by the bone that controls it, if such a bone exists. The mesh is then transformed relative to the bone transformation.

If you compile and run your code now, you will see your model on the screen! It is a spaceship with detail texture. But if you can resist the urge to compile your project and run, you can easily make the model rotate in real-time so you can see all of it:

- In the code, find the [Update](#) method.
- Modify the code (including adding the lines shown above the method) to look like this:

C#

```

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    modelRotation += (float)gameTime.ElapsedGameTime.TotalMilliseconds *
        MathHelper.ToRadians(0.1f);

    base.Update(gameTime);
}

```

And that's it. Compile and run your project by hitting the F5 key or clicking the **Debug** menu, and then clicking **Start Debugging**.



Congratulations!

You did it. There is a lot to making games, but you have taken the first step. A 3D model with lighting and movement in real time. From here, there is no limit to where you could go!

For simplicity's sake, we took some shortcuts that could be optimized for better performance. An obvious improvement would be to precalculate the [View](#) and [Projection](#) matrices instead of calculating them every time [Draw](#) is called, since they do not change. Try out this optimization as a first step. When you are ready to make your game interactive, go to the next tutorial.

Next...

[Tutorial 2: Making Your Model Move Using Input](#)

Ideas to Expand

Got the urge to tinker with the project a bit? Try these ideas.

- Modify the lighting parameters in the [Draw](#) call. Look at [BasicEffect](#) for an idea of what you can modify.
- Instead of looking at a blue background, try adding an image as your background. See [How To: Make a Scrolling Background](#) for guidance. Hint: Make sure you use a call to [SpriteBatch.Draw](#) that allows you to specify a *layerDepth* parameter, and set that depth to 1.0f.
- Get more ideas and resources at [XNA Creators Club Online](#).

Tutorial 2: Making Your Model Move Using Input

This article details how to use the XNA Framework Input API to take user input from an Xbox 360 Controller and apply it to the model displayed in Tutorial 1.

For more information on using a keyboard to control the model, see [Optional Step: Controlling the Ship With Keyboard Input](#).

Note

This tutorial builds on code you have written during the previous tutorial: [Tutorial 1: Displaying a 3D Model on the Screen](#). Follow the steps in the previous tutorial before starting this tutorial.

- [The Complete Sample](#)
- [Step 1: Connect Your Xbox 360 Controller](#)
- [Step 2: Create Variables to Turn and Move the Model](#)
- [Step 3: Take Input from the User](#)
- [Optional Step: Controlling the Ship With Keyboard Input](#)
- [Congratulations!](#)
- [Ideas to Expand](#)

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download GoingBeyond2_Tutorial_Sample.zip](#).

Step 1: Connect Your Xbox 360 Controller

The first step in this tutorial is to make sure you can provide some input to your game. We'll use the Xbox 360 Controller. Designed for use with both a Windows computer and an Xbox 360 console, the controller features many analog and digital inputs, as well as vibration motors to give feedback to the user.

This tutorial uses only the Xbox 360 Controller, but there are more ways to take input: the XNA Framework has support for keyboard and mouse devices. Mouse devices are supported only on Windows, not on Xbox 360. For more information on the different input types, see [Input Overview](#).

For now, connect your Xbox 360 Controller and get ready to code!

Step 2: Create Variables to Turn and Move the Model

We want our ship to move around on the screen. To do that, we'll need to create some variables to track the position and orientation of our model in the world.

Fortunately, from our last tutorial ([Tutorial 1: Displaying a 3D Model on the Screen](#)), we have two variables to do just that: *modelPosition*, which is a three-dimensional vector, and *modelRotation*, which is a floating-point value.

Currently, this system allows three degrees of translation (changing position in the world), but only one degree of rotation (changing orientation). For this demonstration, we will use that limitation to simplify our input. In many 3D games, there are three degrees of translation, and three degrees of rotation, but this is a good start.

What we can do right now to make input a little more interesting is add another vector for velocity. By updating the position with the velocity each frame, our 3D model can accelerate and decelerate smoothly. Let's try it.

- Make sure your project from [Tutorial 1: Displaying a 3D Model on the Screen](#) is open. If it isn't, open it by selecting the **File** menu, clicking **Open Project**, and browsing to your project.
- View the code by double-clicking `Game1.cs` in Solution Explorer.
- In the code, find the `Update` method. Modify it to look like this:

C#

```
// Set the velocity of the model, applied each frame to the model's position.  
Vector3 modelVelocity = Vector3.Zero;
```

```

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
        this.Exit();

    // Get some input.
    UpdateInput();

    // Add velocity to the current position.
    modelPosition += modelVelocity;

    // Bleed off velocity over time.
    modelVelocity *= 0.95f;

    base.Update(gameTime);
}

```

The code you've just added to input runs every frame, and does a few different things. First, it gets rid of the code that automatically rotates the ship. You'll be controlling that by using your controller. Next, it calls a method named **UpdateInput**. That method does not exist yet. You'll have to create it in the next step. Last, it adds the velocity of our model to its position, moving it in the world by its velocity, and decays the velocity so that eventually the model will slow down.

Step 3: Take Input from the User

Now that the model is set up to move with velocity, you now need to provide some logic that will change the velocity based on controller input.

A simple system that we can use is an orientation thrust method: in essence, you can point the front of your model in different directions using your controller's thumbstick, then apply thrust in the direction you are pointing using your controller's trigger. By building up thrust in a direction, the model will begin to move.

We can map our controls to the game this way.

- Pressing left or right on the left thumbstick will increase or decrease the value of the *modelRotation* variable by some amount.
- Pressing on the right trigger will add a vector in the direction of our *modelRotation* variable to our *modelVelocity* vector.
- Pressing the **A** button will reset the position, velocity, and rotation values of the model to "warp" the ship back to the center of the screen.

Note

Both the triggers and the thumbsticks are analog controls, meaning that they can report their movements in varying amounts, rather than just on or off. On the Xbox 360 Controller, the two thumbsticks and two triggers are analog controls, and all other buttons are digital buttons.

Let's code it!

- Find some empty space in your code below the `Update` method.
- Add a new method called **protected void UpdateInput()**.
- Modify the method to look like this:

C#

```

protected void UpdateInput()
{
    // Get the game pad state.
    GamePadState currentState = GamePad.GetState(PlayerIndex.One);
    if (currentState.IsConnected)
    {
        // Rotate the model using the left thumbstick, and scale it down
        modelRotation -= currentState.ThumbSticks.Left.X * 0.10f;
    }
}

```

```

    // Create some velocity if the right trigger is down.
    Vector3 modelVelocityAdd = Vector3.Zero;

    // Find out what direction we should be thrusting,
    // using rotation.
    modelVelocityAdd.X = -(float)Math.Sin(modelRotation);
    modelVelocityAdd.Z = -(float)Math.Cos(modelRotation);

    // Now scale our direction by how hard the trigger is down.
    modelVelocityAdd *= currentState.Triggers.Right;

    // Finally, add this vector to our velocity.
    modelVelocity += modelVelocityAdd;

    GamePad.SetVibration(PlayerIndex.One,
        currentState.Triggers.Right,
        currentState.Triggers.Right);

    // In case you get lost, press A to warp back to the center.
    if (currentState.Buttons.A == ButtonState.Pressed)
    {
        modelPosition = Vector3.Zero;
        modelVelocity = Vector3.Zero;
        modelRotation = 0.0f;
    }
}
}
}

```

That method does a lot. Let's take it piece by piece to investigate exactly what you're doing with input and the model.

```

// Get the game pad state.
GamePadState currentState = GamePad.GetState(PlayerIndex.One);

```

This call to [GetState](#) retrieves a [GamePadState](#) object, which contains the information we need about the controller—in this case, thumbstick and trigger positions.

```

// Rotate the model using the left thumbstick, and scale it down
modelRotation -= currentState.ThumbSticks.Left.X * 0.10f;

```

Retrieving the x-axis value of the left thumbstick (left and right movement) returns a value that is added to the *modelrotation* variable. The value is scaled down so that the rotation isn't too fast.

```

// Create some velocity if the right trigger is down.
Vector3 modelVelocityAdd = Vector3.Zero;

// Find out what direction we should be thrusting,
// using rotation.
modelVelocityAdd.X = -(float)Math.Sin(modelRotation);
modelVelocityAdd.Z = -(float)Math.Cos(modelRotation);

// Now scale our direction by how hard the trigger is down.
modelVelocityAdd *= currentState.Triggers.Right;

```

A little math here helps translate the rotation of the ship into a vector. Taking the sine value of the rotation gives us the proper amount of X (left and right) movement, and the cosine gives us the Z (forward and back) movement. Then, we take the vector and lengthen it by how hard the player is holding down the right trigger.

```

// Finally, add this vector to our velocity.
modelVelocity += modelVelocityAdd;

```

Finally, the created vector is added to the current velocity vector to create the final velocity vector that will be applied to move the model around.


```
GamePad.SetVibration(PlayerIndex.One,
    currentState.Triggers.Right,
    currentState.Triggers.Right);
```

We're using the right trigger values to give some feedback to the player with the Xbox 360 Controller vibration motors, using [SetVibration](#). The Xbox 360 Controller has two motors that run at different speeds, so experiment to find the best combination for the action that's happening in the game.

```
// In case you get lost, press A to warp back to the center.
if (currentState.Buttons.A == ButtonState.Pressed)
{
    modelPosition = Vector3.Zero;
    modelVelocity = Vector3.Zero;
    modelRotation = 0.0f;
}
```

This little extra will move the model back to its original position and orientation in case it leaves the screen.

Optional Step: Controlling the Ship With Keyboard Input

An Xbox controller is not the only means of control available for your ship. You can also use the standard keyboard that is hooked up to your computer. The following instructions assume that you have completed the previous steps in this tutorial.

⚠Caution

Keyboard input is not supported for the Xbox 360 platform. Controlling the ship on an Xbox 360 requires an Xbox 360 Controller.

The keyboard-specific code will be added to the `Game1.Update` and `Game1.UpdateInput` methods that were modified earlier. Let's handle the easy stuff first – modification of the `Game1.Update` method.

Modifying the Update Method

Currently, you can only exit your game with the controller or by clicking the **Close** button. Let's modify the following code:

```
if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
```

to match this:

```
if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed ||
    Keyboard.GetState().IsKeyDown(Keys.Escape))
```

The new code now closes the game when you press the Escape key.

Modifying the UpdateInput Method

This step involves more code than the previous step. Once you have modified the code, you can control the ship's rotation with the A (rotate left) and D (rotate Right) keys and accelerate with the Enter key.

In the `Game1.UpdateInput` method, after the existing declaration of the `currentState` variable, add the following line of code:

```
KeyboardState currentKeyState = Keyboard.GetState();
```

This captures the current state of the keyboard, including which keys were being pressed at the time.

Remove the following line (Don't forget the matching curly brace!):

```
if (currentState.IsConnected)
{
```

This allows keyboard input to be received if a controller is not connected.

Search farther down the code and find the line that assigns the X value of the left thumbstick to `modelRotation`. Right before that line, add the following:

```
if (currentKeyState.IsKeyDown(Keys.A))
    modelRotation += 0.10f;
else if (currentKeyState.IsKeyDown(Keys.D))
    modelRotation -= 0.10f;
```

```
else
```

This first examines the keyboard for input (specifically, the A and D keys) and changes the rotation value accordingly. If no keyboard input is detected, the game controller is checked for input.

Search farther down for the following line of code:

```
modelVelocityAdd *= currentState.Triggers.Right;
```

Right before that code, add the following:

```
if (currentKeyState.IsKeyDown(Keys.W))
    modelVelocityAdd *= 1;
else
```

This code checks to see if the W key is pressed and, if so, increases the current velocity. If the W key is not pressed, the game controller is checked for input.

The final modification warps the ship back to its original position if the Enter key is pressed.

Look for the code that checks the state of the A button on the controller (near the end of the method). Modify this line to also check for a key press. After the modification, it should look like this:

```
if (currentState.Buttons.A == ButtonState.Pressed || currentKeyState.IsKeyDown(Keys.Enter))
```

Rebuild and run the game and you can now steer the ship with the A and D keys, accelerate with the W key, and warp back with the Enter key!

Congratulations!

At this point, your ship moves and gives you feedback through your Xbox 360 Controller. The player is in control of the action.

When you're ready, let's add the final element—audio—to get you on your way. Once the player can control the action, and see and hear the results of their actions, you're well on your way to creating a game.

Next...

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Ideas to Expand

Want to play around some more with input? Try these ideas.

- Change the game to view your model from the top, as in a top-down arcade game. (Hint: Play with the *cameraPosition* vector. Note that you can't set it exactly up and down because the camera vector cannot be the same as the "up" vector.)
- Scale the vibration to occur more powerfully as the ship approaches the viewer. (Hint: Use the distance between *modelPosition* and *cameraPosition*.)
- Try using a keyboard to control the ship. See the [Keyboard](#) class. (Hint: You can plug a USB keyboard into your Xbox 360 console.)
- Get more ideas and resources at [XNA Creators Club Online](#).

Tutorial 3: Making Sounds with XNA Game Studio

Details how to use the XNA Framework Audio API to play them.

Note

This tutorial builds on code you have written during the previous tutorial: [Tutorial 2: Making Your Model Move Using Input](#). Follow the steps in the previous tutorial before you start this tutorial.

- [The Complete Sample](#)
- [Step 1: Get Some Wave Files](#)
- [Step 2: Loading the Wave Files by Using the Content Pipeline](#)
- [Step 3: Play Sounds Using the Audio API](#)
- [Congratulations!](#)

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download GoingBeyond3_Tutorial_Sample.zip](#).

Step 1: Get Some Wave Files

Audio in XNA Game Studio is wave based. There are two methods for using wave files in your game: using XACT to create and play your sound effects or using the [SoundEffect](#) class. We'll use the second method because it lets you quickly get some cool sound effects into your game.

Note

For more information on using XACT, see [Audio Overview](#).

The first thing to do is get some wave files. The files used in this tutorial are included in the (GoingBeyond3_Tutorial_Sample.zip). Download this file now and extract it to its own directory.

- Make sure your project from [Tutorial 2: Making Your Model Move Using Input](#) is open. If it is not, open it by clicking the **File** menu, and then click **Open Project** and browse to your project.
- In Solution Explorer, right-click the **Content** node, click **Add**, and then click **New Folder**. Name this folder `Audio`.
- Right-click the **Audio** folder you just created, click **Add**, and then click **New Folder**. Name this folder `Waves`.
- Open Windows Explorer. Browse to the folder containing the content from your recently extracted sample file and from there to the `Content\Audio\Waves` folder. Inside that folder, you will see two audio files: `engine_2.wav` and `hyperspace_activate.wav`.
- Copy those files to the `Content\Audio\Waves` folder in Windows Explorer.
- Right-click the **Waves** folder, click **Add**, and then click **Existing Item**. Using the dialog box that appears, browse back to the location of the extracted tutorial sample files and find the `Contents\Audio\Waves` folder. Select both wave files, and click **OK**.

Step 2: Loading the Wave Files by Using the Content Pipeline

You probably remember that you have already used the content pipeline to load content into your game. In this step, you will use that same technique to load sound effects into your game. With a few simple lines of code, you can play those sound effects at the appropriate time.

Open your `Game1.cs` file, and take a look at the **LoadContent** method. This is where the ship model is loaded by the content pipeline. You will add new code that declares two [SoundEffect](#) members and one [SoundEffectInstance](#), and loads the recently added sound effect files into those new member variables.

- After the code that declares the **myModel** class member, add the following code:

C#

```
//Set the sound effects to use
SoundEffect soundEngine;
SoundEffectInstance soundEngineInstance;
```

```
SoundEffect soundHyperspaceActivation;
```

- Modify the **LoadContent** code (including adding the lines shown above the method) to look like this:

C#

```
// The aspect ratio determines how to scale 3d to 2d projection.
float aspectRatio;

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    myModel = Content.Load<Model>("Models\\p1_wedge");
    soundEngine = Content.Load<SoundEffect>("Audio\\Waves\\engine_2");
    soundEngineInstance = soundEngine.CreateInstance();
    soundHyperspaceActivation =
        Content.Load<SoundEffect>("Audio\\Waves\\hyperspace_activate");

    aspectRatio = graphics.GraphicsDevice.Viewport.AspectRatio;
}
```

In these steps, you declared two member variables, and told the content pipeline to load two wave files into your game when **LoadContent** is called at the beginning of your game. You also assigned a **SoundEffectInstance** to one of those files, so you can monitor the playback later.

The code now loads the wave files. Your next step is to play them at the appropriate times.

Step 3: Play Sounds Using the Audio API

You can access the sounds you wish to play in your game through a **SoundEffect** object, or play directly by calling **Play**.

For your looping engine sound, call **SoundEffectInstance.Play** to begin looping the engine sound effect. Use the **SoundEffectInstance** to pause and resume the sound effect as your engines turn on and off when the user holds the trigger. When the player presses the **A** button to warp, play the hyperspace sound by calling **SoundEffect.Play**. Calling **SoundEffect.Play** plays the sound, but it doesn't allow us to pause it (which is okay for the hyperspace sound, but not for the engine sound).

- Find the **UpdateInput** method. Modify it to look like this:

C#

```
protected void UpdateInput()
{
    // Get the game pad state.
    GamePadState currentState = GamePad.GetState(PlayerIndex.One);
    if (currentState.IsConnected)
    {
        // Rotate the model using the left thumbstick, and scale it down
        modelRotation -= currentState.ThumbSticks.Left.X * 0.10f;

        // Create some velocity if the right trigger is down.
        Vector3 modelVelocityAdd = Vector3.Zero;

        // Find out what direction we should be thrusting,
        // using rotation.
        modelVelocityAdd.X = -(float)Math.Sin(modelRotation);
        modelVelocityAdd.Z = -(float)Math.Cos(modelRotation);

        // Now scale our direction by how hard the trigger is down.
        modelVelocityAdd *= currentState.Triggers.Right;

        // Finally, add this vector to our velocity.
    }
}
```

```

        modelVelocity += modelVelocityAdd;

        GamePad.SetVibration(PlayerIndex.One,
            currentState.Triggers.Right,
            currentState.Triggers.Right);

        //Play engine sound only when the engine is on.
        if (currentState.Triggers.Right > 0)
        {

            if (soundEngineInstance.State == SoundState.Stopped)
            {
                soundEngineInstance.Volume = 0.75f;
                soundEngineInstance.IsLooped = true;
                soundEngineInstance.Play();
            }
            else
                soundEngineInstance.Resume();
        }
        else if (currentState.Triggers.Right == 0)
        {
            if (soundEngineInstance.State == SoundState.Playing)
                soundEngineInstance.Pause();
        }

        // In case you get lost, press A to warp back to the center.
        if (currentState.Buttons.A == ButtonState.Pressed)
        {
            modelPosition = Vector3.Zero;
            modelVelocity = Vector3.Zero;
            modelRotation = 0.0f;
            soundHyperspaceActivation.Play();
        }
    }
}

```

Many things are happening here. Here is a breakdown of what you are doing.

```
SoundEffectInstance soundEngineInstance;
```

The [SoundEffectInstance](#) represents an instance of a sound. In this case, **soundEngineInstance** will represent the sound of your engines when you hold the right trigger.

```

//Play engine sound only when the engine is on.
if (currentState.Triggers.Right > 0)
{

    if (soundEngineInstance.State == SoundState.Stopped)
    {
        soundEngineInstance.Volume = 0.75f;
        soundEngineInstance.IsLooped = true;
        soundEngineInstance.Play();
    }
    else
        soundEngineInstance.Resume();
}
else if (currentState.Triggers.Right == 0)
{
    if (soundEngineInstance.State == SoundState.Playing)
        soundEngineInstance.Pause();
}

```

This code manages the engine sound. Since you enter this code once each frame, you have to make sure you do not

continually try to play the same sound. You only want to modify the state of the [SoundEffectInstance](#) if there is a change, such as the trigger being released after being held. If the sound is stopped (which it will be at the start of the game), we call [Play](#) to start it up.

From that point forward, each release of the trigger will call [Pause](#) and halt playback of the [SoundEffectInstance](#). Subsequently, holding the trigger again will call [Resume](#), and playback will continue.

```
// In case you get lost, press A to warp back to the center.
if (currentState.Buttons.A == ButtonState.Pressed)
{
    modelPosition = Vector3.Zero;
    modelVelocity = Vector3.Zero;
    modelRotation = 0.0f;
    soundHyperspaceActivation.Play();
}
```

Finally, pressing the **A** button warps, gets, and plays a sound all at once using [Play](#). Since you do not need to stop or pause this sound, but can just let it play, there is no reason to hold on to the sound in a [SoundEffectInstance](#) object.

Congratulations!

At this point, you have a spaceship floating in 3D space. This spaceship moves around when you use your Xbox 360 Controller, it makes sounds, and it gives you feedback in your controller. You have created the very beginnings of a 3D game using XNA Game Studio, and you are only just getting started. There is so much more to explore!

Next...

[Tutorial 4: Make a Game in 60 Minutes](#)

Ideas to Expand

If you are ready to go further with this sample, why not try a few of these ideas?

- Use some of the advanced features of the [Play](#) to change the volume and pitch of your engines as you change pressure on your right trigger.
- Add some background music and try setting different volumes for sound effects and background music.
- Get more ideas and resources at [XNA Creators Club Online](#).

Tutorial 4: Make a Game in 60 Minutes

This tutorial helps you learn about the process of game construction, while guiding you through writing a relatively complete game.

Note

This tutorial builds on code you have written during the previous tutorial: [Tutorial 3: Making Sounds with XNA Game Studio](#). Follow the steps in the previous tutorial before starting this tutorial.

- [The Complete Sample](#)
- [Introduction](#)
- [Before You Begin: Getting the Project Ready](#)
- [Step 1: Ship Shape](#)
- [Step 2: Camera Work](#)
- [Step 3: You Need Rocks. Lots of Them.](#)
- [Step 4: When Ships and Asteroids Collide](#)
- [Step 5: Boom - You're Dead](#)
- [Step 6: Revenge of the Ship](#)
- [Step 7: Space, the Final Frontier](#)
- [Finally](#)

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download GoingBeyond4_Tutorial_Sample.zip](#).

Introduction

Understanding the basics of game coding is the single most-difficult step for a beginning game programmer. While it is easy to find samples that show completed games, tips and tricks, or tutorials that show you how to do specific techniques, there are very few that help you through the process of game construction. The objective of this tutorial is to help you learn about the process of game construction, while guiding you through writing a relatively complete game. In addition, this tutorial will use only those assets found in the complete sample file ([GoingBeyond4_Tutorial_Sample.zip](#)), eliminating the need to install additional content. Download the sample file now and extract its contents to a directory on your local drive.

The game you implement will be a simple clone of the popular Asteroids® game by Atari®. The place of Asteroids in video game history is well known, and you are encouraged to read the [interesting history of the game on Wikipedia](#). This tutorial assumes you have a general idea of how the Asteroids game works.

A lot of initial work in this tutorial is already done for you. In fact, this tutorial picks up at the end of the [Tutorial 3: Making Sounds with XNA Game Studio](#) tutorial. Once you have completed the first three tutorials in [Going Beyond: XNA Game Studio in 3D](#), you will have a moveable spaceship with sounds and rendering in 3D space. In another 60 to 90 minutes of coding time, you will have a relatively complete Asteroids-style game.

Before You Begin: Getting the Project Ready

Begin this tutorial by completing the first three tutorials in the [Going Beyond: XNA Game Studio in 3D](#) series, or by downloading the completed code for the third tutorial (Video Tutorial 3: Making Sounds with XNA Game Studio and XACT) from the [XNA Creators Club Online Web site](#).

Step 1: Ship Shape

The first three tutorials in the [Going Beyond: XNA Game Studio in 3D](#) series explained the basics of a single interactive object, rendered in 3D. A true game, however, needs more than just one object. The first step toward making this tutorial into a game is to prepare the game to track and render several objects.

Think of the idea of your ship on the screen. It is drawn using a `Model` class, it has a position tracked by a `Vector3`, and still another `Vector3` tracks velocity. A `float` tracks the rotation angle. Each of these data types is modified or checked in different places along the code path, and while the end result looks good to the user, the drawback comes when you try to extend the gameplay to include another object that needs similar data.

If, for instance, you wanted to add a second ship that would also draw on the screen, and had the ability to move and turn, you would have to create a copy of each of the variables you were using for the first ship. You would have to duplicate the code you wrote that checked and modified each variable. Each copied line would be nearly identical to the original line, except that it was acting on a new variable.

For a game that will ultimately have more than a dozen objects all drawing and moving around, this is unworkable. The duplicated code would make your code unreadable and painful to modify. However, there is a better way. If you create a code object that holds the common variables that allow you to draw and move a 3D object, then maintain a list of these objects, you can draw and move them all together using the same code. This process is called encapsulation, and is the beginning of object-oriented programming, which becomes more and more important the larger your game becomes.

Start by right-clicking on your project in Solution Explorer, and select **Add**, then **Class**. Type `Ship.cs` into the **Name** box, then click **Add**.

When you add the new file, it will open up in the code window. This new file represents a class, or code object. This particular class is named `Ship`. You will notice it is very minimal now; modify it so it looks like the following:

```
C#  
  
using System;  
using System.Collections.Generic;  
using System.Text;  
using Microsoft.Xna.Framework;  
using Microsoft.Xna.Framework.Content;  
using Microsoft.Xna.Framework.Graphics;  
using Microsoft.Xna.Framework.Input;  
  
namespace GoingBeyond4  
{  
    class Ship  
    {  
        public Model Model;  
        public Matrix[] Transforms;  
  
        //Position of the model in world space  
        public Vector3 Position = Vector3.Zero;  
  
        //Velocity of the model, applied each frame to the model's position  
        public Vector3 Velocity = Vector3.Zero;  
  
        public Matrix RotationMatrix = Matrix.Identity;  
        private float rotation;  
        public float Rotation  
        {  
            get { return rotation; }  
            set  
            {  
                float newVal = value;  
                while (newVal >= MathHelper.TwoPi)  
                {  
                    newVal -= MathHelper.TwoPi;  
                }  
                while (newVal < 0)  
                {  
                    newVal += MathHelper.TwoPi;  
                }  
  
                if (rotation != newVal)  
                {  
                    rotation = newVal;  
                    RotationMatrix = Matrix.CreateRotationY(rotation);  
                }  
            }  
        }  
    }  
  
    public void Update(GamePadState controllerState)  
    {  
        // Rotate the model using the left thumbstick, and scale it down.
```



```

        Rotation -= controllerState.ThumbSticks.Left.X * 0.10f;

        // Finally, add this vector to our velocity.
        Velocity += RotationMatrix.Forward * 1.0f *
            controllerState.Triggers.Right;
    }
}

```

You can see that the `Ship` class now does a lot—it holds onto the ship's position, velocity, rotation, and 3D model, and has its own `Update` method that will move the ship around.

Now that you have created the `Ship` class, you need to change the code in the `Game1.cs` code file to take advantage of this new, encapsulated data. Double-click on **Game1.cs** in your Solution Explorer.

Start with drawing the ship's model. Your original drawing code was inside the `Draw` method, but that will not scale up to multiple objects very well. You will be drawing `Model` objects on the screen, so create a method that will draw a chosen `Model`. Below the `Draw` method, add a new method called `DrawModel`, like so:

C#

```

public static void DrawModel(Model model, Matrix modelTransform,
    Matrix[] absoluteBoneTransforms)
{
    //Draw the model, a model can have multiple meshes, so loop
    foreach (ModelMesh mesh in model.Meshes)
    {
        //This is where the mesh orientation is set
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.World =
                absoluteBoneTransforms[mesh.ParentBone.Index] *
                modelTransform;
        }
        //Draw the mesh, will use the effects set above.
        mesh.Draw();
    }
}

```

This `DrawModel` method takes your model-drawing algorithm and applies it to any `Model` object passed into it, drawing the `Model` on the screen. Next, modify the `Draw` call so that it calls this new method:

C#

```

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    Matrix shipTransformMatrix = ship.RotationMatrix
        * Matrix.CreateTranslation(ship.Position);
    DrawModel(ship.Model, shipTransformMatrix, ship.Transformations);
    base.Draw(gameTime);
}

```

The code from the previous tutorial contained declarations for `modelPosition` and `modelRotation` values above the `Draw` call. Delete those—you will not need them anymore. Also delete the `cameraPosition` variable—you will recreate this later.

Next, modify the `Update` and `UpdateInput` methods to use the values in the new `Ship` class as follows:

C#

```

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();
}

```

```

// Get some input.
UpdateInput();

// Add velocity to the current position.
ship.Position += ship.Velocity;

// Bleed off velocity over time.
ship.Velocity *= 0.95f;

base.Update(gameTime);
}

protected void UpdateInput()
{
    // Get the game pad state.
    GamePadState currentState = GamePad.GetState(PlayerIndex.One);
    if (currentState.IsConnected)
    {
        ship.Update(currentState);

        //Play engine sound only when the engine is on.
        if (currentState.Triggers.Right > 0)
        {
            if (soundEngineInstance.State == SoundState.Stopped)
            {
                soundEngineInstance.Volume = 0.75f;
                soundEngineInstance.IsLooped = true;
                soundEngineInstance.Play();
            }
            else
                soundEngineInstance.Resume();
        }
        else if (currentState.Triggers.Right == 0)
        {
            if (soundEngineInstance.State == SoundState.Playing)
                soundEngineInstance.Pause();
        }

        // In case you get lost, press A to warp back to the center.
        if (currentState.Buttons.A == ButtonState.Pressed)
        {
            ship.Position = Vector3.Zero;
            ship.Velocity = Vector3.Zero;
            ship.Rotation = 0.0f;
            soundHyperspaceActivation.Play();
        }
    }
}
}

```

Above the `UpdateInput` method, remove the `modelVelocity` variable above `Update`—it is no longer needed.

Finally, you need to make a change to the way your initialization and content loading are handled. Starting from the top of the `Game` class and continuing down to just above the call to `Update`, modify the code as follows:

C#

```

GraphicsDeviceManager graphics;

//Camera/View information
Vector3 cameraPosition = new Vector3(0.0f, 0.0f, -5000.0f);
Matrix projectionMatrix;
Matrix viewMatrix;

//Audio Components
SoundEffect soundEngine;
SoundEffectInstance soundEngineInstance;

```

```

SoundEffect soundHyperspaceActivation;

//Visual components
Ship ship = new Ship();

public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
}

/// <summary>
/// Allows the game to perform any initialization it needs to before
/// starting to run. This is where it can query for any required
/// services and load any non-graphic related content.
/// Calling base.Initialize will enumerate through any components
/// and initialize them as well.
/// </summary>
protected override void Initialize()
{
    projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.ToRadians(45.0f),
        GraphicsDevice.DisplayMode.AspectRatio,
        1.0f, 10000.0f);
    viewMatrix = Matrix.CreateLookAt(cameraPosition,
        Vector3.Zero, Vector3.Up);

    base.Initialize();
}

private Matrix[] SetupEffectDefaults(Model myModel)
{
    Matrix[] absoluteTransforms = new Matrix[myModel.Bones.Count];
    myModel.CopyAbsoluteBoneTransformsTo(absoluteTransforms);

    foreach (ModelMesh mesh in myModel.Meshes)
    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.EnableDefaultLighting();
            effect.Projection = projectionMatrix;
            effect.View = viewMatrix;
        }
    }
    return absoluteTransforms;
}

protected override void LoadContent()
{
    ship.Model = Content.Load<Model>("Models/p1_wedge");
    ship.Transform = SetupEffectDefaults(ship.Model);
    soundEngine = Content.Load<SoundEffect>("Audio/Waves/engine_2");
    soundEngineInstance = soundEngine.CreateInstance();
    soundHyperspaceActivation =
        Content.Load<SoundEffect>("Audio/Waves/hyperspace_activate");
}

/// <summary>
/// UnloadContent will be called once per game and is the place to unload
/// all content.
/// </summary>
protected override void UnloadContent()
{
}

```

While it may seem like a lot of work, the modified code is a good example of encapsulation, and will come in handy as you develop your game.

Step 2: Camera Work

Now that you have a ship object ready, the next step is to get the ship flying around the screen from a top-down point of view. You will accomplish this by simply changing the camera's angle and distance. Finally, you will adjust the rotation mechanics on the user input, so that it matches the behavior you want.

Reverse the camera position along the z-axis by simply altering the value from negative 5000 to positive 25000. The `cameraPosition` member is declared near the start of the `Game1` class. Now your `cameraPosition` declaration will look like this:

C#

```
Vector3 cameraPosition = new Vector3(0.0f, 0.0f, 25000.0f);
```

Unfortunately, if you run the tutorial with only that change, the ship does not show up. This is because the "projection matrix" of the camera is not correct. The formal term that describes the problem is "bounding frustum culling" (also called "viewing frustum culling"). Look in the XNA Game Studio documentation for the [BoundingFrustum](#) class, which includes a key diagram to help you learn more about frustums and how they relate to the camera. A camera's near and far plane is set in a specific way to (usually) address performance concerns. In this case, the camera's original near plane is 1 and the far plane is at 10,000. When the camera was set at 5,000 units, like in Figure 1, the ship was in the camera's view space.

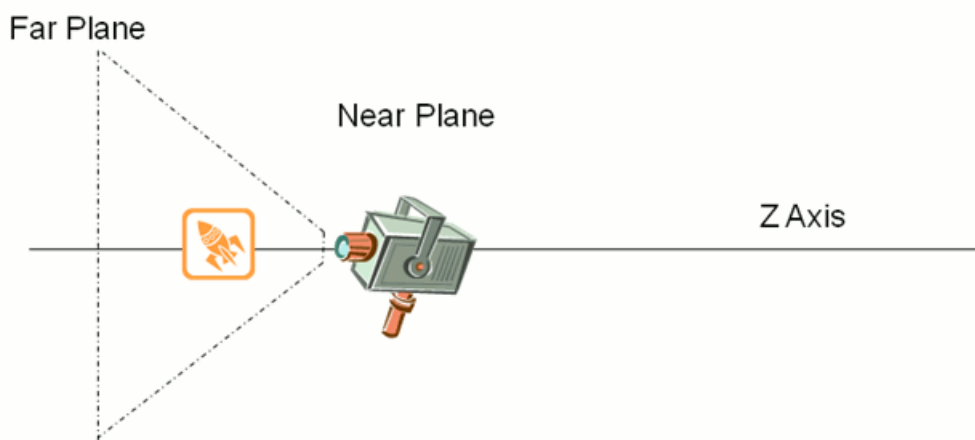


Figure 1. Original camera setting and view space

That's perfectly fine when the spaceship was located 5,000 units away from the camera. But when you moved the camera starting point to 25,000, the camera's view space was in the wrong place, as in Figure 2, leaving the ship too far away to be seen.



Figure 2. New camera position with incorrect view space

Correct the viewing space problem now. Inside the `Initialize` method of the `Game1` class, you will see the method that creates the `projectionMatrix`:

C#

```
projectionMatrix = Matrix.CreatePerspectiveFieldOfView(  
    MathHelper.ToRadians(45.0f),  
    GraphicsDevice.DisplayMode.AspectRatio,  
    1.0f, 10000.0f);
```

You need to change the near and far clipping planes of the frustum so that the ship is back in the viewing space. You determine

the near and far clipping planes by simple math. The camera is 25,000 units away from the ship, so you set the near plane 5,000 units "closer" to the camera, and the far plane 5,000 units "farther away," like this:

C#

```
projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.ToRadians(45.0f),
    GraphicsDevice.DisplayMode.AspectRatio,
    20000.0f, 30000.0f);
```

This corrects the viewing space so that the ship is inside it, as in Figure 3.

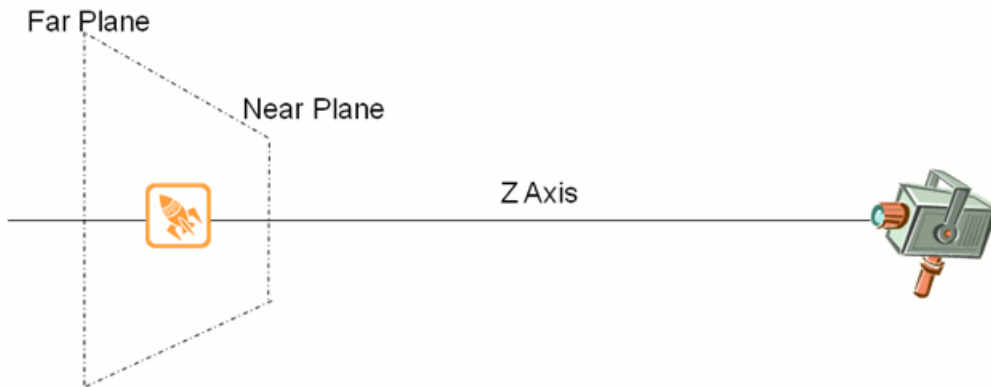


Figure 3. Corrected viewing space

Running the program now will show you the view facing the back end of the ship, rather than facing the nose end. If you fly the ship toward or away from you, you will see the ship disappear as it moves outside the frustum after a few seconds. Now alter the orientation of the ship and how it responds to your input.

In the `Ship` class, you will change the default orientation of the ship so that it starts from a "top-down" perspective. Double-click on **Ship.cs** in Solution Explorer.

If the ship is initially facing away from you, a 90-degree rotation along the x-axis will give you the top-down view. Don't forget, you are looking at the ship down the z-axis, so from your perspective, changes in x are "left/right" and changes in y are "up/down." Thus, rotating the ship on the x-axis flips the ship around as if it were spinning on the wings. In the XNA Framework, angular measurements are given in radians, which means you are rotating the ship $\text{Pi}/2$ radians.

Replace the existing declaration of `RotationMatrix` with the following:

C#

```
public Matrix RotationMatrix =
    Matrix.CreateRotationX(MathHelper.PiOver2);
```

Now, every time you change the ship's rotation (in the ship's `Rotation` property "set" method), you alter the rotation matrix to include this default rotation, plus the rotation amount along the z-axis supplied by the player's controller. You could just as well rotate along any other axis, provided you:

- Position the camera properly.
- Perform your translation and rotation calculations in relation to the correct axis.

Failing to properly calculate translation and rotation movement can yield some surprising, if not frustrating, results. Avoid this by modifying the set method of the `Rotation` property. Change the existing `if` clause to match the following:

C#

```
if (rotation != value)
{
    rotation = value;
    RotationMatrix =
        Matrix.CreateRotationX(MathHelper.PiOver2) *
        Matrix.CreateRotationZ(rotation);
}
```

You should notice that your ship appears to be flying slowly now. That is because your view is much farther away than it used

to be. Just under the declaration of `Velocity`, add a floating-point constant that you can use to adjust the ship's velocity:

C#

```
//amplifies controller speed input
private const float VelocityScale = 5.0f;
```

At the end of the ship's `Update` method, change the current `Velocity` computation to use the `VelocityScale` value to give the ship a little extra speed (more accurately, it increases the number of units per frame in the game):

C#

```
Velocity += RotationMatrix.Forward * VelocityScale *
    controllerState.Triggers.Right;
```

Running with these changes will now give you a top-down view of the ship, which you can fly around on the screen. If you fly off the screen, press the **warp** button. It's a good idea to change the original use of the **A** button to another button, as you will be using the **A** button to fire in a later step.

Step 3: You Need Rocks. Lots of Them.

You have a ship in the game, so now add asteroids to it. For the sake of simplicity, you are only going to track each asteroid's position, direction, and speed. Create a simple class that has only those three members. Right-click on the **GoingBeyond4Windows** project in Solution Explorer, click **Add**, and then click **Class**. Name it `Asteroid.cs`. (Don't forget to add a using statement for `Microsoft.Xna.Framework`). Because this class is "lightweight," you will change it from a class to a structure. (Literally, change the word "class" to "struct" in the file.) There are many nuances about when to use and not use a structure (called a "value type" in C# parlance), which are beyond the scope of this document. Many of the issues relate to performance and garbage collection (GC). In a blog post by the Compact Framework team (<http://blogs.msdn.com/netcfteam/archive/2006/12/22/managed-code-performance-on-xbox-360-for-xna-part-2-gc-and-tools.aspx>) they say this about value types:

"Games typically have lots of small objects that represent game state. The obvious optimization here is to reduce live object count. You can do that by defining those data structures as structs which are value types (to use more general terminology). Value types stay off the GC heap... of course that assumes that your structs don't get boxed in to objects, which can often happen unknowingly in your code."

In this case, you will use a value type for the `Asteroid` (and later for bullets) to reduce garbage collection events, as well as to keep the implementation simple.

Add these three members to the structure:

C#

```
public Vector3 position;
public Vector3 direction;
public float speed;
```

Double-click on your `Game1.cs` file. Inside your `Game1` class, you will create a simple array that contains asteroids. Add some additional members to your `Game1` class to render the asteroids. After the declaration for the ship (`Ship ship = new Ship();`), add the following:

C#

```
Model asteroidModel;
Matrix[] asteroidTransforms;
Asteroid[] asteroidList = new Asteroid[GameConstants.NumAsteroids];
Random random = new Random();
```

There is something new in each of these four lines, so look at each one. The first line is an object that holds on to a lot of information that describes the actual asteroid model loaded by the Content Pipeline processor. You will do that shortly. The second line retains state information related to specific lighting and effect transformations on the asteroid. Because you are not adding any special lighting effects, you will set up a default effect on the model and leave it. The third line is a simple array of asteroids, but you will notice the introduction of the `GameConstants` class, which will generally hold values that you might want to change as you develop and test the game. There will be more about that shortly. The final line creates a random number generator, which you will use for a few purposes in the game.

Look at this new `GameConstants` class briefly. One nice design trick for simple games like this is to gather game parameters, which you might want to customize, into a single location. Create that class now. Click **Add**, and then click **Class**. Name it `GameConstants.cs`. Once the file opens, add these constants to the class (you will use the `PlayfieldSize` constants later):

C#

```
//camera constants
public const float CameraHeight = 25000.0f;
public const float PlayfieldSizeX = 16000f;
public const float PlayfieldSizeY = 12500f;
//asteroid constants
public const int NumAsteroids = 10;
```

As you might guess from the addition of the camera constants, you will want to modify the `CameraPosition` declaration in the `Game1` class to look like this now:

C#

```
Vector3 cameraPosition = new Vector3(0.0f, 0.0f,
    GameConstants.CameraHeight);
```

And the initialization of the `projectionMatrix` (located in the `Initialize` method) to look like this:

C#

```
projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.ToRadians(45.0f),
    GraphicsDevice.DisplayMode.AspectRatio,
    GameConstants.CameraHeight - 1000.0f,
    GameConstants.CameraHeight + 1000.0f);
```

Turn your attention back to the `Asteroid` structure again. To render the asteroid, you need to add an asteroid model to the Content Pipeline. You already have a `Content/Models` directory in your game, since it is storing your ship model. Add the "asteroid1.x" model to that directory by right-clicking on the directory, clicking **Add**, and then clicking **Existing Item**. Then navigate back to the path to which you extracted the contents of the sample file. (Remember, you had to do this when you did "Tutorial 1: Displaying a 3D Model on the Screen" from the "Going Beyond: XNA Game Studio in 3D" series). Select the **asteroid1.x** file from your `Content/Models` directory (you might need to select files of type "Content Pipeline Files" to see it), and add it to your `Models` directory. In addition to adding this model, you will also need to manually copy the asteroid's texture file, "asteroid1.tga," from the `Content/Textures` directory of the sample to the `Content/Textures` subfolder in your game project folder. Just manually copy it. Do not use the **Add**, and then **Existing Item** approach. Also, be very careful about the copying process. A common beginner's mistake is to copy a Texture file into a Model directory. This is not a good idea.

Now you will visit the `LoadContent` method in the `Game1` class. This is where you will load the mesh model for your asteroid that you just added. Just below the line where you added the `p1_wedge` model, load the asteroid model and transforms:

C#

```
asteroidModel = Content.Load<Model>("Models/asteroid1");
asteroidTransforms = SetupEffectDefaults(asteroidModel);
```

Next, you'll need a method to populate the `asteroidList` with several asteroids. It will be called at the end of the `Initialize` method in the `Game1` class (before the `base.Initialize()` call). When you create an asteroid, you will give it a starting speed and random direction. For now, start the asteroids from the center of the screen.

Create a separate method called `ResetAsteroids`, which will populate the list of asteroids.

```
private void ResetAsteroids()
{
    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        asteroidList[i].position = Vector3.Zero;
        double angle = random.NextDouble() * 2 * Math.PI;
        asteroidList[i].direction.X = -(float)Math.Sin(angle);
        asteroidList[i].direction.Y = (float)Math.Cos(angle);
        asteroidList[i].speed = GameConstants.AsteroidMinSpeed +
```

```
        (float)random.NextDouble() * GameConstants.AsteroidMaxSpeed;
    }
}
```

Note

You will need to add two floating-point constants (code given below), `AsteroidMinSpeed` and `AsteroidMaxSpeed`, to the `GameConstants` class yourself. In this example, 100.0 is the minimum speed, and 300.0 is the maximum.

C#

```
public const float AsteroidMinSpeed = 100.0f;
public const float AsteroidMaxSpeed = 300.0f;
```

Then add a call to `ResetAsteroids()` just before the call to `base.Initialize()` in the `Initialize` method.

The direction values of the asteroids are using a basic trigonometric function to determine the x and y components of the direction, based on the starting angle. Do not modify the z value because the game only plays in two dimensions.

Now that you have created the asteroids, you need to render them. You should recognize that this should go in the `Draw()` method. Indeed, you will simply look through the `asteroidList` and render each asteroid in the same manner as the ship. So, add this code after the completion of the rendering of the ship in the `Draw()` method.

C#

```
for (int i = 0; i < GameConstants.NumAsteroids; i++)
{
    Matrix asteroidTransform =
        Matrix.CreateTranslation(asteroidList[i].position);
    DrawModel(asteroidModel, asteroidTransform, asteroidTransforms);
}
```

If you run this code as-is right now, you will see the ship and single asteroid rendered in the center. There are actually 10 asteroids there, but they are stacked one on top of the other.

The next step is to give the asteroids some motion. This is accomplished in the `Update()` method by simply iterating over the list and updating their position. Do that just after you update the ship's velocity:

C#

```
for (int i = 0; i < GameConstants.NumAsteroids; i++)
{
    asteroidList[i].Update(timeDelta);
}
```

One thing you added is a time delta. This is a small efficiency trick. Calculate the `timeDelta` value once per update, rather than repeatedly calling the property to check for the total seconds passed. This will be the first line of the `Update()` method (in the `Game1` class):

C#

```
float timeDelta = (float)gameTime.ElapsedGameTime.TotalSeconds;
```

Notice that you are calling each asteroid's `Update()` method in this loop, so you will need to add that method to the `Asteroid` structure (inside the structure's braces, not outside it). Thanks to the expressiveness of the XNA Framework Math library, this can be written in a very simple manner:

```
public void Update(float delta)
{
    position += direction * speed * GameConstants.AsteroidSpeedAdjustment * delta;
}
```

Note

You will need to add the floating-point constant, `AsteroidSpeedAdjustment`, to the `GameConstants` class. In this case, use a default value of 5.0.

If everything went well, you will see the asteroids all flying away from the ship in random directions until they all disappear from the screen.

What is wrong with this picture?

Keep the asteroids in the game by wrapping the asteroid around the screen. This is accomplished by allowing the asteroids to drift off the screen, then shifting them to the other side once they have disappeared. The values of playfield size constants were made from some rough approximations based on the actual viewing space. A properly designed game will carefully calculate the field of view area and determine the limits based on asteroid model sizes, and other parameters. In this case, use the `PlayfieldSize` constants in a simple fashion to determine the "wraparound" trigger areas. After you update the asteroid's position in the `Asteroid` class's `Update` method, you then determine if you need to move the asteroid around:

C#

```
if (position.X > GameConstants.PlayfieldSizeX)
    position.X -= 2 * GameConstants.PlayfieldSizeX;
if (position.X < -GameConstants.PlayfieldSizeX)
    position.X += 2 * GameConstants.PlayfieldSizeX;
if (position.Y > GameConstants.PlayfieldSizeY)
    position.Y -= 2 * GameConstants.PlayfieldSizeY;
if (position.Y < -GameConstants.PlayfieldSizeY)
    position.Y += 2 * GameConstants.PlayfieldSizeY;
```

Now you should see your asteroids calmly wrapping around the screen as they drift through space. Perfect! Well, almost. This game will not be very interesting if you start all the asteroids in the center, since that would result in a collision with the ship. You need to add some code to start the asteroids on the left or right edge of the screen.

Choosing where to start the asteroid is a little tricky. For the x value of the asteroid's position, you must first choose to start on the left or right side of the screen. Use the random number generator to pick either a 0 or 1. If it's 0, you will start on the left. If it's 1, you will start on the right. To do this, call `random.Next(2)`, which generates a number between 0 and up to, but not including, the passed value (so it only returns a 0 or 1). For the y value of the asteroid's position, simply choose a random number that is within the playfield's y range. This means you will modify the line that assigns the asteroid position a value of `Vector3.Zero` so that the final method looks like this:

C#

```
private void ResetAsteroids()
{
    float xStart;
    float yStart;
    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        if (random.Next(2) == 0)
        {
            xStart = (float)-GameConstants.PlayfieldSizeX;
        }
        else
        {
            xStart = (float)GameConstants.PlayfieldSizeX;
        }
        yStart =
            (float)random.NextDouble() * GameConstants.PlayfieldSizeY;
        asteroidList[i].position = new Vector3(xStart, yStart, 0.0f);
        double angle = random.NextDouble() * 2 * Math.PI;
        asteroidList[i].direction.X = -(float)Math.Sin(angle);
        asteroidList[i].direction.Y = (float)Math.Cos(angle);
        asteroidList[i].speed = GameConstants.AsteroidMinSpeed +
            (float)random.NextDouble() * GameConstants.AsteroidMaxSpeed;
    }
}
```

Note

Do not forget to declare the two floating-point values, `xStart` and `yStart`, just before the **for** loop.

It might look a little confusing, but run through the math calculations a couple times to get comfortable with what is going on. At this point, you have a ship in the center of the screen, with several asteroids starting on the sides, moving in random directions and speeds.

Step 4: When Ships and Asteroids Collide

Now, add a few more content items to your game, which you will use in Steps 3 and 4. You will add one model and three sounds to the game:

1. Add "pea_proj.x" (the bullet model) to the Models section in your project. To do this, right-click **Models**, click **Add**, and then click **Existing Item**. Do not forget you might need to change the **Files of Type** drop-down to **Content Pipeline Files**.

The model is located in the downloaded samples directory under Content/Models—in the same place the asteroid model was lurking. You will also need to copy the "pea_proj.tga" file from the Content/Texture location to your Content/Textures location. Again, do not use **Add** and then **Existing Item** here.

2. As in Tutorial 3 ("Making Sounds with XNA Game Studio"), navigate to the Content/Audio/Waves directory of the downloaded sample directory and copy weapons/explosion3.wav, explosions/explosion2.wav, and weapons/tx0_fire1.wav into your Content/Audio/Waves directory.

Right after the existing sound effect variables, add three new ones to store the sound effects you just added:

C#

```
SoundEffect soundExplosion2;  
SoundEffect soundExplosion3;  
SoundEffect soundWeaponsFire;
```

Now, modify the `LoadContent` method to load your new sound effects:

C#

```
soundExplosion2 =  
    Content.Load<SoundEffect>("Audio/Waves/explosion2");  
soundExplosion3 =  
    Content.Load<SoundEffect>("Audio/Waves/explosion3");  
soundWeaponsFire =  
    Content.Load<SoundEffect>("Audio/Waves/tx0_fire1");
```

The new explosion and weapons fire effects are now ready to be used when needed during game play.

You have something visually interesting now. You have a ship, with sound effects, that you can move around. You also have asteroids happily flying around on the screen. Unfortunately, you cannot shoot the asteroids. On the other hand, the asteroids also cannot hurt you—yet. Add some collision detection between the ship and the asteroids. In a later step, you will get even by shooting back.

With the XNA Framework, simple collision detection is easy. In this step, you will be using a [BoundingSphere](#), which is an object that creates the smallest-sized sphere (by default) that can enclose the target model. The **BoundingSphere** contains many different intersection tests, including the ability to detect intersections with planes, rays, boxes, and, of course, other spheres (among other things). Hence, you will put an invisible bubble around each object you want to test, and then determine if they intersect each other.

One trick to remember in gameplay is that you should consider different rules for collisions, depending on the context. In this case, you will deliberately create a bounding sphere around the ship that is smaller than the ship. This is a little game programming trick. Most models are uneven in shape, but a **BoundingSphere** only takes into account the point farthest from the model's center when creating the sphere's radius. This results in collisions that often appear like they were nowhere near the player's ship. In addition, creating a slightly smaller sphere gives a little more "forgiveness" in case a player gets too close to an asteroid. So, create two constants in the `GameConstants` class that sets bounding sphere sizes for the asteroids and ship:

C#

```
public const float AsteroidBoundingSphereScale = 0.95f; //95% size  
public const float ShipBoundingSphereScale = 0.5f; //50% size
```

Now, create the actual bounding sphere around the ship, just after you update the asteroid positions in the `Update` method of

the `Game1` class. Then create a loop that visits each asteroid. Inside this loop, you create a temporary bounding sphere around the asteroid and determine whether the ship and asteroid sphere are intersecting. If the two spheres intersect, you play an explosion sound and break out of the loop:

C#

```
//ship-asteroid collision check
BoundingSphere shipSphere = new BoundingSphere(
    ship.Position, ship.Model.Meshes[0].BoundingSphere.Radius *
        GameConstants.ShipBoundingSphereScale);
for (int i = 0; i < asteroidList.Length; i++)
{
    BoundingSphere b = new BoundingSphere(asteroidList[i].position,
        asteroidModel.Meshes[0].BoundingSphere.Radius *
            GameConstants.AsteroidBoundingSphereScale);
    if (b.Intersects(shipSphere))
    {
        //blow up ship
        soundExplosion3.Play();
        break; //exit the loop
    }
}
```

Running this program now gives you some great feedback. First, the collision check seems to work pretty well. Second, you hear a collision sound. Third, the sound does not seem right. This is because as the asteroid and ship move through each other, the collision check is constantly firing every frame, with the XNA Framework trying to play the explosion in every frame, causing garbled sound. You can solve this problem by removing the colliding objects from the updating and rendering. In a real game, this means the ship explodes and you lose a life. In the tutorial, simply remove the ship and the offending asteroid from the display and then update. You'll add this feature in the next step.

Step 5: Boom - You're Dead

The code starts getting a little more complex now, but you will leverage some handy secrets in XNA Game Studio to make it easy. To start, you need to create a Boolean flag that tells you if the ship is alive or dead. This will go in the `Ship` class, right after the declaration of `VelocityScale`:

```
public bool isActive = true;
```

Before you test for a ship and asteroid collision, you need to verify the `isActive` flag is **true**. This is done by wrapping the collision code you already wrote in an **if** statement. This is easy with XNA Game Studio. Highlight the entire block of code that does the collision check (the **BoundingSphere** declaration and the loop right after it), then right-click the selected code and click **Surround With**, then select the **if** statement (not the **#if** statement) from the list. You will see your code is now wrapped in an **if** statement, awaiting a Boolean condition. Now all you have to do is replace **true** with `ship.isActive`. Finally, set `ship.isActive` to **false** after you play the explosion sound.

This fixes the explosion sound, but both the ship and the offending asteroid are still visible in the game. First, remove the ship. Since you have set the flag in the `Update()` method, that still leaves you the responsibility to not draw the ship anymore. So once again wrap a chunk of code in the `Draw()` method with the **if** statement. By now you should be familiar with what portion of the code draws the ship. Select the line of code that draws the ship, right-click, click **Surround With**, and insert an `if (ship.isActive)` test.

Running the code now should let you merrily smash your ship into an asteroid, with an accompanying explosion and the disappearance of your ship.

Finally, you need to remove the colliding asteroid. This requires a flag just like the ship. Each asteroid needs an `isActive` flag that tells whether you should draw or update the asteroid. This is accomplished in five steps, which you should attempt to do on your own:

1. Create an `isActive` flag inside the `Asteroid` class, similar to what you did with the ship.
2. Set the `isActive` flag to **true** when you create each asteroid in the `ResetAsteroids` method in the `Game1` class.
3. In the code where you draw the asteroids, surround the drawing code in an **if** statement. This happens inside the loop where you iterate through each asteroid.
4. Similarly, you now need to do the same thing in the update section, checking to see whether an asteroid is active before you execute a collision test with the ship.

5. If a ship does collide with an asteroid, set that asteroid's active state to **false** just after you play the explosion sound.

If you did all the steps correctly, you should have an almost-functional game! Collisions, sounds, moving ships. It's all starting to come together! This leads to the next question: What to do once you blow up the ship? Easy. Press the **Warp** button. In [Tutorial 2: Making Your Model Move Using Input](#), you wrote some code that reset the ship back to the center. It's still there and still useful (except back then it was the **A** button). Now go ahead and add a `ship.isActive = true` statement in the code block for pressing the **B** button. (Hint: Look in the `UpdateInput` method in the `Game1` class). Also, if you have not changed the Warp button from **A** to **B**, now is the time to do it. Instant life-regeneration!

The next step will add bullets to the game, so you can shoot back. The good news is that all the work you have done up to now will make the bullet work seem easy.

Step 6: Revenge of the Ship

In many ways, a bullet in the game is like an asteroid: it travels in a direction and collides with things. You are going to treat bullets just a little differently though, giving the game a little fine-tuning in the process.

Conveniently, the `Bullet` structure is exactly like the `Asteroid` structure, so all you need to do is copy the `Asteroid` implementation file, rename the new file to `Bullet.cs`, and the structure name to `Bullet`. In addition, you will want to add these new constants to the `GameConstants` class for later use:

C#

```
public const int NumBullets = 30;
public const float BulletSpeedAdjustment = 100.0f;
```

Think ahead a little bit right now, though. How long do you want the bullets to fly around in space? Do you want them to wrap around the screen? Maybe only live for a certain number of seconds or travel a certain distance? Do you want the bullets to be able to collide with both asteroids and the ship? Any of these approaches are legitimate ways to make the game physics behave. In this case, though, the bullets are simply going to disappear once they go off the screen. This means the `Update()` method in the `Bullet` class will flag the bullet as inactive once it drifts off the view.

This is a simple check, similar to what was done with the `Asteroid` structure:

C#

```
public void Update(float delta)
{
    position += direction * speed *
        GameConstants.BulletSpeedAdjustment * delta;
    if (position.X > GameConstants.PlayfieldSizeX ||
        position.X < -GameConstants.PlayfieldSizeX ||
        position.Y > GameConstants.PlayfieldSizeY ||
        position.Y < -GameConstants.PlayfieldSizeY)
        isActive = false;
}
```

As with the `Asteroid` structure, you are now done with the `Bullet` structure. However, you have to do several things to make the bullets actually work in the game. You have done this all before with the asteroids, but review the basic steps:

1. Load the model into the Content Pipeline and set the effect transforms.
2. Create a list to track all bullets in the game.
3. Create a bullet and make a firing sound when a player presses a specific button.
4. Draw the bullet in-flight.
5. Test the asteroids and bullets for collisions. If they collide, make an explosion sound and remove the colliding bullet and asteroid.

Begin by creating the needed instance variables. Underneath the same place that you created the `asteroidList` and `asteroidModel` variables, create a list to hold the bullets and a model to hold the bullet's shape.

C#

```
Model bulletModel;
Matrix[] bulletTransforms;
Bullet[] bulletList = new Bullet[GameConstants.NumBullets];
```

Then in the `LoadContent()` method, assign the `pea_proj` model to `bulletModel`. Remember, you added `pea_proj.x` to the `Content/Models` directory earlier:

C#

```
bulletModel = Content.Load<Model>("Models/pea_proj");
bulletTransforms = SetupEffectDefaults(bulletModel);
```

Unlike the asteroids, you do not create bullets inside `Initialize`. Instead, create a bullet every time a user presses the **A** button on the controller. Add a new condition to the `UpdateInput()` method at the very end:

C#

```
//are we shooting?
if (ship.isActive && currentState.Buttons.A == ButtonState.Pressed)
{
    //add another bullet. Find an inactive bullet slot and use it
    //if all bullets slots are used, ignore the user input
    for (int i = 0; i < GameConstants.NumBullets; i++)
    {
        if (!bulletList[i].isActive)
        {
            bulletList[i].direction = ship.RotationMatrix.Forward;
            bulletList[i].speed = GameConstants.BulletSpeedAdjustment;
            bulletList[i].position = ship.Position + (200 * bulletList[i].direction);
            bulletList[i].isActive = true;
            soundWeaponsFire.Play();
            score -= GameConstants.ShotPenalty;
            break; //exit the loop
        }
    }
}
```

There is an interesting trick in the above code that needs explaining. When it calculate the initial position of the bullet, it appears as if it's firing out the nose of the ship. Thus, the code begins by determining where the bullet is starting from, which is the ship's center. Then it translates the bullet 200 additional units in the direction of the bullet (200 is the rough approximation of the distance from the ship's center to the nose of the ship).

This kind of "motion offset" is very common in game development. One "extra credit" feature you can do is to add the ship's current velocity to the bullet's velocity.

Now it's actually possible to run your game and press the fire (**A**) button, but you will not yet be able to see the bullets (because you have not drawn them). When you press the fire button (the **A** button), you might have observed that the sound behaves just like the original problem you had with the asteroid/ship explosions. You are triggering the sound too many times. In fact, you probably noticed that you can hold the fire button down and it will fire a continuous "stream" of bullets (until all the bullet "slots" are used). There is a simple fix to the `UpdateInput()` method to fire the bullet only once every time the button is pressed.

The problem with `UpdateInput` is that it is failing to track the user's previous input state. Create a variable that does this. Just after the `GraphicsDeviceManager` declaration (near the beginning of the `Game1` class), add this variable:

C#

```
GamePadState lastState = GamePad.GetState(PlayerIndex.One);
```

Then, at the end of the `UpdateInput` method, save the user's game pad state:

C#

```
lastState = currentState;
```

Now all you need to do is change the `if` statement for the "fire" effect to verify that the button was not held down the last time the code updated:

```
if (ship.isActive && currentState.Buttons.A == ButtonState.Pressed &&
```

```
lastState.Buttons.A == ButtonState.Released)
```

When you run the program, you will now hear a firing sound for every time you individually press the **A** button. Now that you see how to do this, add the same check to your **hyperspace** button for consistency reasons. The next step is to draw the bullet as it is flying around the screen. Conveniently, this code is identical to the code that draws the asteroids, except you replace the word "asteroid" with "bullet" (in the `Draw` method):

C#

```
for (int i = 0; i < GameConstants.NumBullets; i++)
{
    if (bulletList[i].isActive)
    {
        Matrix bulletTransform =
            Matrix.CreateTranslation(bulletList[i].position);
        DrawModel(bulletModel, bulletTransform, bulletTransforms);
    }
}
```

Then you will again do exactly the same thing in the `Update` method. Just after the part where you update the asteroid positions (but before you do the asteroid/ship collision test), add the code to update the bullets:

C#

```
for (int i = 0; i < GameConstants.NumBullets; i++)
{
    if (bulletList[i].isActive)
    {
        bulletList[i].Update(timeDelta);
    }
}
```

If you run the code at this point in time, you actually have an "almost working" game! All that is left is testing for collisions between the bullet and the asteroids. This process is really quite easy. All you need to do is loop through each asteroid, checking to see if a bullet is colliding with it. If so, deactivate both the colliding bullet and asteroid and continue through the list of asteroids until you are done. The code is almost literally a copy of the ship/asteroid collision code, except instead of `if (shipAlive)` you have a loop through each asteroid. One thing to note: Do this collision check before checking to see if the ship collides with an asteroid—that way, the player gets credit for a "kill" before getting destroyed!

C#

```
//bullet-asteroid collision check
for (int i = 0; i < asteroidList.Length; i++)
{
    if (asteroidList[i].isActive)
    {
        BoundingBox asteroidSphere =
            new BoundingBox(asteroidList[i].position,
                asteroidModel.Meshes[0].BoundingBox.Radius *
                    GameConstants.AsteroidBoundingBoxScale);
        for (int j = 0; j < bulletList.Length; j++)
        {
            if (bulletList[j].isActive)
            {
                BoundingBox bulletSphere = new BoundingBox(
                    bulletList[j].position,
                    bulletModel.Meshes[0].BoundingBox.Radius);
                if (asteroidSphere.Intersects(bulletSphere))
                {
                    soundExplosion2.Play();
                    asteroidList[i].isActive = false;
                    bulletList[j].isActive = false;
                    break; //no need to check other bullets
                }
            }
        }
    }
}
```

```
}  
}
```

If everything went well, you can now fly a ship around, shoot asteroids, and collide with asteroids. Congratulations, you have written your first XNA Framework game! But wait, the blue background looks, well, nothing at all like a good Asteroids game. You need a space background and, of course, a way to keep score. That is the last step.

Step 7: Space, the Final Frontier

The last step will be to add finishing touches to the game to make it both visually appealing and to give it more of a game feel. You will do this in two parts. The first part is to add a 2D background texture to the game to give it a nice space appearance. The second part will be adding a simple scoring mechanism to the game. When it comes to doing either step, the first thing to remember is that all 2D items are drawn as sprites. A background and score are no different in terms of how they are drawn, but as you will learn, it does matter when they are drawn.

For the first step, you need to create a texture for the starry background. Begin by adding the `stars Texture2D` object in the same place you declared your `Asteroid` and `Bullet` models:

C#

```
Texture2D stars;
```

Just after you create the `bulletModel` and `bulletTransforms` objects, load the texture:

C#

```
stars = Content.Load<Texture2D>("Textures/B1_stars");
```

Lastly, at the beginning of the `Draw()` method, just after you call `Clear` on the graphics device, draw the star background. It's important to draw the background at the beginning instead of the end, otherwise, it will obscure everything already drawn (asteroids, and so on) by laying the background on top of the previously drawn objects.

C#

```
spriteBatch.Begin(SpriteBlendMode.None, SpriteSortMode.Immediate,  
    SaveStateMode.None);  
spriteBatch.Draw(stars, new Rectangle(0, 0, 800, 600), Color.White);  
spriteBatch.End();
```

Now add the `B1_stars.tga` file into the `Content/Textures` area in your project (right-click **Textures**, click **Add**, and then click **Existing Item**). Then browse to the `Content\Textures` folder of the extracted sample and select the `B1_stars.tga` file. When you run your game now, you should see a pretty star field in the background, with all your gameplay in the foreground.

All that is left is keeping score in the game. This is accomplished in a few simple steps:

1. Create a sprite font and add it to the Content Pipeline processing.
2. Load the sprite font with the rest of your content.
3. Set the display string and call the `DrawString` method.

Creating the sprite font is simple. The first thing to do is create a new folder under the `Content` folder called `Fonts`. Then right-click this folder, click **Add**, and then click **New Item**. From the menu, pick `Sprite Font`. The default file name for this file is `SpriteFont1.spritefont`. While you could leave it that way, give it the same name as the font you want to use. Since you will be using the `Kootenay` font, name the file `Kootenay.spritefont`. Feel free to experiment with different fonts later, once you are comfortable with this process. Once you create the file, it will open to allow you to edit the different font parameters. Just accept the settings and close it for now.

Note

Before you go on, it's important to understand that fonts are very technical pieces of art. The people and companies that create them pour an enormous amount of work in them. In many cases, fonts are protected under copyright and licensing terms that widely vary. Just because a font is installed on your computer does not mean you automatically have the right to redistribute the font to anybody else. Keep this in mind if you ever decide to share games that you write. Fortunately, the default font used by the `Sprite Font` item is redistributable. For more information, see [How To: Draw Text](#).

Now that you created the sprite font, add some code in the `Game1` class so that you can display something. Just after you declare the `stars` object, add a few more declarations:

C#

```
SpriteFont kootenay;
int score;
Vector2 scorePosition = new Vector2(100, 50);
```

The first declaration will hold the sprite font. The second is a simple counter for the score. Finally, the `scorePosition` object will let you position the score in screen coordinates. You could just as well move the `scorePosition` into the `GameConstants` class, but due to compilation rules regarding the `Vector2` class, you cannot make it a `const` value.

Loading the sprite font is a one-line addition to the end of the `LoadContent` method:

C#

```
kootenay = Content.Load<SpriteFont>("Fonts/Kootenay");
```

All that is left is to display the score on the screen. This is pretty simple, provided you respect the rules of drawing order. So far, there are four very distinct drawing steps in the `Draw` method. Draw the background and then the game elements (ship, then asteroids, then bullets). As mentioned previously, if you draw the background after the game elements, all you see is the star field, because drawing the star field last covers the entire screen space. This same issue applies for the game score. Draw the game score last so that it appears overlaid on the rest of the game.

Hopefully by now, you will realize that the score will be drawn just before the `base.Draw` call is made in the `Draw` method. The actual code to draw the string is simply a sprite batch `Begin/End` pair, with the call to `DrawString` in between:

C#

```
spriteBatch.Begin(SpriteBlendMode.AlphaBlend,
    SpriteSortMode.Immediate, SaveStateMode.None);
spriteBatch.DrawString(kootenay, "Score: " + score,
    scorePosition, Color.LightGreen);
spriteBatch.End();
```

When you run the game now, you should see a score displayed in the upper-left corner. You will also notice that the game elements appear to render underneath the score, giving the effect you want. Now think about how you want to score the game.

Good gameplay not just about "running and gunning," it's about forcing the player to make decisions and tradeoffs to achieve one or more goals. In this game, you are going to penalize the player for each round the player fires (offensive actions come at a cost) and presses the **Warp** button (defensive actions come at a cost). You will also penalize the player for dying. In most video games, you are given a limited number of lives, and you subtract a "life" when the player's avatar gets destroyed. However, in this game, a multi-life system is not implemented (you should do that as "extra credit"), so simply take away points. Also, reward points for each asteroid destroyed. First, set up some scoring values in the `GameConstants` class:

C#

```
public const int ShotPenalty = 1;
public const int DeathPenalty = 100;
public const int WarpPenalty = 50;
public const int KillBonus = 25;
```

Now, alter the scores in the appropriate places. For instance, the shot penalty would be added to the `UpdateInput` method, just after it registers that the player fired a bullet, most likely just after the `soundWeaponsFire.Play()` line:

```
score -= GameConstants.ShotPenalty;
```

You will need similar approaches in three other areas, which you should accomplish on your own:

- When the ship is determined to have collided with an asteroid (subtract `DeathPenalty` from score).
- When a bullet is determined to have collided with an asteroid (add `KillBonus` to score).
- When the player presses the warp button (subtract `WarpPenalty` from score).

Finally

The initial goal of this tutorial was to show you that the tools, materials, and knowledge to write a game are right at your fingertips, and to guide you through the process of writing your first game. By now, you have learned how to:

- Change camera views to achieve different rendering perspectives.
- Write simple collision-detection routines.
- Create a game environment where many things appear to be happening at once.
- Integrate 2D and 3D rendering.
- Render text in your game.
- Create a feel of "gameplay" where the player has both benefits and penalties with their decisions.

Hopefully, you have also enjoyed the process of making the game. After all, making a game should be just as much fun as playing one! But this is only the beginning. While the game you made is interesting, there are *many* things you can still do to make the game more engaging and enjoyable. Here are several suggestions (but by no means a complete list) on how you can take your game to the next level:

- Wrap the ship around on the screen.
- Vibrate the controller when a ship collides with an asteroid.
- Split the big asteroids into successively smaller ones.
- Add explosion effects when a bullet hits an asteroid.
- Add engine particle effects as the ship flies around.
- Add a smart "UFO" that attacks the player's ship.
- Add a "high score" capability to the game.
- Determine when the playing field is cleared and start a new level, perhaps with more or faster asteroids.

At this point, you've been given many of the basic elements you need to build a game: graphics, input, and sound. Even so, you may be wondering, "How do I build a game?"

Games are an expressive process, with plenty of room for creative problem solving. There is truly no one right way to make a game. With the example you have created, there are still many missing elements. What else does the ship interact with? Does it have a goal? What obstacles prevent the ship from reaching the goal?

Answering these questions will define your game, and make it your own. Play some games that inspire you, check out the [XNA Creators Club Online](#), read up on the [Programming Guide](#), explore the XNA Framework, and have fun building a game of your very own. We hope you enjoy XNA Game Studio!

Tutorial 5: Adding Multiplayer and Networking Support to the Game

This tutorial adds two-player competitive game play to the game completed in Tutorial 4.

Note

In order to test the network functionality of this sample, each instance must be running on a separate computer, each with XNA Game Studio installed.

- [Before You Begin: Getting the Project Ready](#)
- [Step 1: Simplify the Update Method](#)
- [Step 2: Encapsulate Player Code](#)
- [Step 3: Split the Screen](#)
- [Step 4: Game State Management](#)
- [Step 5: Receive Network Data](#)



Before You Begin: Getting the Project Ready

Begin this tutorial by completing the fourth tutorial in the [Going Beyond: XNA Game Studio in 3D](#) series, or by downloading the completed code for the fourth tutorial (Video Tutorial 4: Make a Game in 60 Minutes).

[Download GoingBeyond4_Tutorial_Sample.zip.](#)

A complete code sample for this tutorial is also available for you to download, including full source code and any additional supporting files required by the sample.

[Download GoingBeyond5_Tutorial_Sample.zip.](#)

Step 1: Simplify the Update Method

This tutorial is going to add some details to the **Update** and **Draw** methods of your game, so as a first step we will work on simplifying the code in these methods.

Move the bullet-asteroid collision check into a method to simplify the code.

C#

```
bool CheckForBulletAsteroidCollision(float bulletRadius,  
float asteroidRadius)
```

```

{
    for (int i = 0; i < asteroidList.Length; i++)
    {
        if (asteroidList[i].isActive)
        {
            BoundingSphere asteroidSphere =
                new BoundingSphere(asteroidList[i].position,
                    asteroidRadius *
                    GameConstants.AsteroidBoundingSphereScale);
            for (int j = 0; j < bulletList.Length; j++)
            {
                if (bulletList[j].isActive)
                {
                    BoundingSphere bulletSphere =
                        new BoundingSphere(bulletList[j].position,
                            bulletRadius);
                    if (asteroidSphere.Intersects(bulletSphere))
                    {
                        asteroidList[i].isActive = false;
                        bulletList[j].isActive = false;
                        score += GameConstants.KillBonus;
                        return true; //no need to check other bullets
                    }
                }
            }
        }
    }
    return false;
}

```

Similarly, we will make new methods for the ship-asteroid collision check, and replace the corresponding code in the **Update** method with the call to this new **CheckForShipAsteroidCollision** method.

C#

```

public bool CheckForShipAsteroidCollision(float shipRadius,
    float asteroidRadius)
{
    //ship-asteroid collision check
    if (ship.isActive)
    {
        BoundingSphere shipSphere = new BoundingSphere(ship.Position,
            shipRadius * GameConstants.ShipBoundingSphereScale);
        for (int i = 0; i < asteroidList.Length; i++)
        {
            if (asteroidList[i].isActive)
            {
                BoundingSphere b =
                    new BoundingSphere(asteroidList[i].position,
                        asteroidRadius *
                        GameConstants.AsteroidBoundingSphereScale);
                if (b.Intersects(shipSphere))
                {
                    //blow up ship
                    //soundExplosion3.Play();
                    ship.isActive = false;
                    asteroidList[i].isActive = false;
                    score -= GameConstants.DeathPenalty;
                    return true;
                }
            }
        }
    }
    return false;
}

```

This class uses the ship model. While you perform this, add a **shipModel** to the **Game** class, and initialize it as you did the

other models. When we add a player class to the code, the players will all share the same ship model data.

C#

```
Model asteroidModel;
Model bulletModel;
Model shipModel;
Matrix[] asteroidTransforms;
Matrix[] bulletTransforms;
Matrix[] shipTransforms;
```

Initialize the ship model with the other models in the game class.

C#

```
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    shipModel = Content.Load<Model>("Models/p1_wedge");
    shipTransforms = SetupEffectDefaults(shipModel);
    asteroidModel = Content.Load<Model>("Models/asteroid1");
    asteroidTransforms = SetupEffectDefaults(asteroidModel);
    bulletModel = Content.Load<Model>("Models/pea_proj");
    bulletTransforms = SetupEffectDefaults(bulletModel);
    stars = Content.Load<Texture2D>("Textures/B1_stars");
    soundEngine = Content.Load<SoundEffect>("Audio/Waves/engine_2");
    soundEngineInstance = soundEngine.CreateInstance();
    soundHyperspaceActivation =
        Content.Load<SoundEffect>("Audio/Waves/hyperspace_activate");
    soundExplosion2 =
        Content.Load<SoundEffect>("Audio/Waves/explosion2");
    soundExplosion3 =
        Content.Load<SoundEffect>("Audio/Waves/explosion3");
    soundWeaponsFire =
        Content.Load<SoundEffect>("Audio/Waves/tx0_fire1");
    lucidaConsole = Content.Load<SpriteFont>("Fonts/Lucida Console");
}
```

The **Update** method should now look something like this:

C#

```
protected override void Update(GameTime gameTime)
{
    float timeDelta = (float)gameTime.ElapsedGameTime.TotalSeconds;

    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    // Get some input.
    UpdateInput();

    // Add velocity to the current position.
    ship.Position += ship.Velocity;

    // Bleed off velocity over time.
    ship.Velocity *= 0.95f;

    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        asteroidList[i].Update(timeDelta);
    }

    for (int i = 0; i < GameConstants.NumBullets; i++)
```

```

    {
        if (bulletList[i].isActive)
        {
            bulletList[i].Update(timeDelta);
        }
    }

    if (CheckForBulletAsteroidCollision(
        bulletModel.Meshes[0].BoundingSphere.Radius,
        asteroidModel.Meshes[0].BoundingSphere.Radius))
    {
        soundExplosion2.Play();
    }

    bool shipDestroyed = CheckForShipAsteroidCollision(
        shipModel.Meshes[0].BoundingSphere.Radius,
        asteroidModel.Meshes[0].BoundingSphere.Radius);
    if (shipDestroyed)
    {
        soundExplosion3.Play();
    }

    base.Update(gameTime);
}

```

Next, we will look for ways to simplify the **UpdateInput** method. The code used to shoot a bullet could be moved into a new method, as well as the code to warp the ship to center and to play the engine sound.

C#

```

public void ShootBullet()
{
    //add another bullet. Find an inactive bullet slot and use it
    //if all bullets slots are used, ignore the user input
    for (int i = 0; i < GameConstants.NumBullets; i++)
    {
        if (!bulletList[i].isActive)
        {
            bulletList[i].direction = ship.RotationMatrix.Forward;
            bulletList[i].speed = GameConstants.BulletSpeedAdjustment;
            bulletList[i].position = ship.Position
                + (200 * bulletList[i].direction);
            bulletList[i].isActive = true;
            score -= GameConstants.ShotPenalty;
            return;
        }
    }
}

```

C#

```

public void WarpToCenter()
{
    ship.Position = Vector3.Zero;
    ship.Velocity = Vector3.Zero;
    ship.Rotation = 0.0f;
    ship.isActive = true;
    score -= GameConstants.WarpPenalty;
}

```

C#

```

void PlayEngineSound(GamePadState currentState)
{
    //Play engine sound only when the engine is on.
    if (currentState.Triggers.Right > 0)

```

```

{
    if (soundEngineInstance.State == SoundState.Stopped)
    {
        soundEngineInstance.Volume = 0.75f;
        soundEngineInstance.IsLooped = true;
        soundEngineInstance.Play();
    }
    else
        soundEngineInstance.Resume();
}
else if (currentState.Triggers.Right == 0)
{
    if (soundEngineInstance.State == SoundState.Playing)
        soundEngineInstance.Pause();
}
}

```

It is also possible to simplify the check for whether a current button is pressed. We will create a new method called **IsButtonPressed** that takes only one parameter, the button to check. This method will check a set of global variables storing the current and last states of the game pad.

C#

```

GamePadState currentState;
GamePadState lastState;

...
bool IsButtonPressed(Buttons button)
{
    switch (button)
    {
        case Buttons.A:
            return (currentState.Buttons.A == ButtonState.Pressed &&
                lastState.Buttons.A == ButtonState.Released);
        case Buttons.B:
            return (currentState.Buttons.B == ButtonState.Pressed &&
                lastState.Buttons.B == ButtonState.Released);
        case Buttons.X:
            return (currentState.Buttons.X == ButtonState.Pressed &&
                lastState.Buttons.X == ButtonState.Released);
        case Buttons.Back:
            return (currentState.Buttons.Back == ButtonState.Pressed &&
                lastState.Buttons.Back == ButtonState.Released);
        case Buttons.DPadDown:
            return (currentState.DPad.Down == ButtonState.Pressed &&
                lastState.DPad.Down == ButtonState.Released);
        case Buttons.DPadUp:
            return (currentState.DPad.Up == ButtonState.Pressed &&
                lastState.DPad.Down == ButtonState.Released);
    }
    return false;
}

```

Using these new methods in **UpdateInput** will result in the following simplified code:

C#

```

protected void UpdateInput()
{
    // Get the game pad state.
    currentState = GamePad.GetState(PlayerIndex.One);
    if (currentState.IsConnected)
    {
        if (ship.isActive)
        {
            ship.Update(currentState);
        }
    }
}

```

```

        PlayEngineSound(currentState);
    }
    // In case you get lost, press B to warp back to the center.
    if (IsButtonPressed(Buttons.B))
    {
        WarpToCenter();
        soundHyperspaceActivation.Play();
    }

    //are we shooting?
    if (ship.isActive && IsButtonPressed(Buttons.A))
    {
        ShootBullet();
        soundWeaponsFire.Play();
        bool isFiring = true;
    }
    lastState = currentState;
}
}

```

Step 2: Encapsulate Player Code

Just as we encapsulated the ship and asteroid data in Tutorial 4, we now need to encapsulate the player data so we can easily create multiple instances of the data needed to save the state of any player in a multiplayer game. In this step, we are going to refactor the code, moving any relevant data and methods from the **Game** class and into a new **Player** class.

The first step is to create a new class to contain the player object code. Right-click on your game project in Solution Explorer, and choose **Add** and then **Class**. In the **Name** field, enter `Player.cs` and click **Add**.

In this class, we are going to use some objects from the **Framework**, **Graphics**, and **Input** namespaces. To do this, add a using statement to the top of the class to include these namespaces.

```

using System;
using System.Collections;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;

namespace GoingBeyond5
{
    public class Player
    {
        public Player()
        {
        }
    }
}

```

Next, look at **Game1.cs** and determine what data should be associated with a player. The player object would keep the information about the game pad state, ship, bullets, asteroids, and score. Move these to `Player.cs`. Upon inspection of the code, you might notice that the random number generator is only used to reset the asteroids for a player, so let's move this to the `Player.cs` file as well. You might also notice that the last gamepad state is something that would be associated with a player.

In addition to this data, we will store a picture for each player as a [Texture2D](#).

C#

```

internal GamePadState lastState;

internal Ship ship = new Ship();
internal Asteroid[] asteroidList =
    new Asteroid[GameConstants.NumAsteroids];
internal Bullet[] bulletList = new Bullet[GameConstants.NumBullets];

internal int score;

Random random = new Random();

```

If you compile your project at this time, you will receive some warnings that these member variables no longer exist in Game1.cs. These warnings can help you determine which methods in Game1.cs deal primarily with the player data.

One of the warnings indicates that the *asteroidList* referenced in the **ResetAsteroids** function does not exist in the current context. Looking at this, you can see that the **ResetAsteroids** function deals with resetting the asteroids for a player. Move the **ResetAsteroids** method into the **Player** class.

The **ResetAsteroids** method was previously called during game initialization, so also move the call to **ResetAsteroids** to the **Player** constructor.

C#

```
using System;
using System.Collections;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;

namespace GoingBeyond5
{
    public class Player
    {
        internal GamePadState lastState;

        internal Ship ship = new Ship();
        internal Asteroid[] asteroidList =
            new Asteroid[GameConstants.NumAsteroids];
        internal Bullet[] bulletList = new Bullet[GameConstants.NumBullets];

        internal int score;

        Random random = new Random();

        public Player()
        {
            ResetAsteroids();
        }

        internal void Update(GameTime gameTime)
        {
            float timeDelta = (float)gameTime.ElapsedGameTime.TotalSeconds;
            // Add velocity to the current position.
            ship.Position += ship.Velocity;

            // Bleed off velocity over time.
            ship.Velocity *= 0.95f;

            for (int i = 0; i < GameConstants.NumAsteroids; i++)
            {
                asteroidList[i].Update(timeDelta);
            }

            for (int i = 0; i < GameConstants.NumBullets; i++)
            {
                if (bulletList[i].isActive)
                {
                    bulletList[i].Update(timeDelta);
                }
            }
        }

        private void ResetAsteroids()
        {
            float xStart;
            float yStart;
            for (int i = 0; i < GameConstants.NumAsteroids; i++)
            {
                if (random.Next(2) == 0)
                {
```



```

        xStart = (float)-GameConstants.PlayfieldSizeX;
    }
    else
    {
        xStart = (float)GameConstants.PlayfieldSizeX;
    }
    yStart =
        (float)random.NextDouble() * GameConstants.PlayfieldSizeY;
    asteroidList[i].position = new Vector3(xStart, yStart, 0.0f);
    double angle = random.NextDouble() * 2 * Math.PI;
    asteroidList[i].direction.X = -(float)Math.Sin(angle);
    asteroidList[i].direction.Y = (float)Math.Cos(angle);
    asteroidList[i].speed = GameConstants.AsteroidMinSpeed +
        (float)random.NextDouble() * GameConstants.AsteroidMaxSpeed;
    asteroidList[i].isActive = true;
    }
}

```

Continue moving methods associated with the player data into the **Player** class. Here, we move the **WarpToCenter**, **ShootBullet**, **CheckForShipAsteroidCollision**, and **CheckForBulletAsteroidCollision** methods into the **Player** class.

There is also some code in the **Update** method that applies specifically to the player. Remove this code from **Game.Update** and create a new method called **Player.Update** with the player specific code. This requires that you add a new instance of the **Player** object to your **Game** class.

C#

```
Player player = new Player();
```

Make these changes to create the new **Update** function in **Player** class.

C#

```

internal void Update(GameTime gameTime)
{
    float timeDelta = (float)gameTime.ElapsedGameTime.TotalSeconds;
    // Add velocity to the current position.
    ship.Position += ship.Velocity;

    // Bleed off velocity over time.
    ship.Velocity *= 0.95f;

    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        asteroidList[i].Update(timeDelta);
    }

    for (int i = 0; i < GameConstants.NumBullets; i++)
    {
        if (bulletList[i].isActive)
        {
            bulletList[i].Update(timeDelta);
        }
    }
}

```

Change the **Update** function in the **Game** to use the new method.

C#

```

protected override void Update(GameTime gameTime)
{
    player.Update(gameTime);

    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();
}

```

```

// Get some input.
UpdateInput();

if (player.CheckForBulletAsteroidCollision(
    bulletModel.Meshes[0].BoundingSphere.Radius,
    asteroidModel.Meshes[0].BoundingSphere.Radius))
{
    soundExplosion2.Play();
}

bool shipDestroyed = player.CheckForShipAsteroidCollision(
    shipModel.Meshes[0].BoundingSphere.Radius,
    asteroidModel.Meshes[0].BoundingSphere.Radius);
if (shipDestroyed)
{
    soundExplosion3.Play();
}

base.Update(gameTime);
}

```

When you are finished, your **Player** class will look like this:

C#

```

using System;
using System.Collections;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;

namespace GoingBeyond5
{
    public class Player
    {
        internal GamePadState lastState;

        internal Ship ship = new Ship();
        internal Asteroid[] asteroidList =
            new Asteroid[GameConstants.NumAsteroids];
        internal Bullet[] bulletList = new Bullet[GameConstants.NumBullets];

        internal int score;

        Random random = new Random();

        public Player()
        {
            ResetAsteroids();
        }

        internal void Update(GameTime gameTime)
        {
            float timeDelta = (float)gameTime.ElapsedGameTime.TotalSeconds;
            // Add velocity to the current position.
            ship.Position += ship.Velocity;

            // Bleed off velocity over time.
            ship.Velocity *= 0.95f;

            for (int i = 0; i < GameConstants.NumAsteroids; i++)
            {
                asteroidList[i].Update(timeDelta);
            }

            for (int i = 0; i < GameConstants.NumBullets; i++)
            {

```

```

        if (bulletList[i].isActive)
        {
            bulletList[i].Update(timeDelta);
        }
    }
}

private void ResetAsteroids()
{
    float xStart;
    float yStart;
    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        if (random.Next(2) == 0)
        {
            xStart = (float)-GameConstants.PlayfieldSizeX;
        }
        else
        {
            xStart = (float)GameConstants.PlayfieldSizeX;
        }
        yStart =
            (float)random.NextDouble() * GameConstants.PlayfieldSizeY;
        asteroidList[i].position = new Vector3(xStart, yStart, 0.0f);
        double angle = random.NextDouble() * 2 * Math.PI;
        asteroidList[i].direction.X = -(float)Math.Sin(angle);
        asteroidList[i].direction.Y = (float)Math.Cos(angle);
        asteroidList[i].speed = GameConstants.AsteroidMinSpeed +
            (float)random.NextDouble() * GameConstants.AsteroidMaxSpeed;
        asteroidList[i].isActive = true;
    }
}

internal void ShootBullet()
{
    //add another bullet. Find an inactive bullet slot and use it
    //if all bullets slots are used, ignore the user input
    for (int i = 0; i < GameConstants.NumBullets; i++)
    {
        if (!bulletList[i].isActive)
        {
            bulletList[i].direction = ship.RotationMatrix.Forward;
            bulletList[i].speed = GameConstants.BulletSpeedAdjustment;
            bulletList[i].position = ship.Position
                + (200 * bulletList[i].direction);
            bulletList[i].isActive = true;
            score -= GameConstants.ShotPenalty;
            return;
        }
    }
}

internal void WarpToCenter()
{
    ship.Position = Vector3.Zero;
    ship.Velocity = Vector3.Zero;
    ship.Rotation = 0.0f;
    ship.isActive = true;
    score -= GameConstants.WarpPenalty;
}

internal bool CheckForBulletAsteroidCollision(float bulletRadius,
float asteroidRadius)
{
    for (int i = 0; i < asteroidList.Length; i++)
    {
        if (asteroidList[i].isActive)
        {

```

```

        BoundingBoxSphere asteroidSphere =
            new BoundingBoxSphere(
                asteroidList[i].position, asteroidRadius *
                GameConstants.AsteroidBoundingBoxSphereScale);
        for (int j = 0; j < bulletList.Length; j++)
        {
            if (bulletList[j].isActive)
            {
                BoundingBoxSphere bulletSphere =
                    new BoundingBoxSphere(bulletList[j].position,
                        bulletRadius);
                if (asteroidSphere.Intersects(bulletSphere))
                {
                    asteroidList[i].isActive = false;
                    bulletList[j].isActive = false;
                    score += GameConstants.KillBonus;
                    return true; //no need to check other bullets
                }
            }
        }
    }
    return false;
}
internal bool CheckForShipAsteroidCollision(float shipRadius,
    float asteroidRadius)
{
    //ship-asteroid collision check
    if (ship.isActive)
    {
        BoundingBoxSphere shipSphere =
            new BoundingBoxSphere(ship.Position, shipRadius *
                GameConstants.ShipBoundingBoxSphereScale);
        for (int i = 0; i < asteroidList.Length; i++)
        {
            if (asteroidList[i].isActive)
            {
                BoundingBoxSphere b =
                    new BoundingBoxSphere(asteroidList[i].position,
                        asteroidRadius *
                        GameConstants.AsteroidBoundingBoxSphereScale);
                if (b.Intersects(shipSphere))
                {
                    //blow up ship
                    //soundExplosion3.Play();
                    ship.isActive = false;
                    asteroidList[i].isActive = false;
                    score -= GameConstants.DeathPenalty;
                    return true;
                }
            }
        }
    }
    return false;
}
}
}
}

```

With this step complete, update the **UpdateInput** method of the game to use the new **Player** class.

C#

```

protected void UpdateInput()
{
    // Get the game pad state.
    currentState = GamePad.GetState(PlayerIndex.One);
    lastState = player.lastState;
}

```

```

if (currentState.IsConnected)
{
    if (player.ship.isActive)
    {
        player.ship.Update(currentState);
        PlayEngineSound(currentState);
    }
    // In case you get lost, press B to warp back to the center.
    if (IsButtonPressed(Buttons.B))
    {
        player.WarpToCenter();
        soundHyperspaceActivation.Play();
    }

    //are we shooting?
    if (player.ship.isActive && IsButtonPressed(Buttons.A))
    {
        player.ShootBullet();
        soundWeaponsFire.Play();
        bool isFiring = true;
    }
    player.lastState = currentState;
}
}

```

Finally, change the **Game.Draw** so that it accesses the data from the new **Player** class.

C#

```

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin(SpriteBlendMode.None, SpriteSortMode.Immediate,
        SaveStateMode.None);
    spriteBatch.Draw(stars, new Rectangle(0, 0, 800, 600), Color.White);
    spriteBatch.End();

    Matrix shipTransformMatrix = player.ship.RotationMatrix
        * Matrix.CreateTranslation(player.ship.Position);
    if (player.ship.isActive)
    {
        DrawModel(shipModel, shipTransformMatrix, shipTransforms);
    }
    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        Matrix asteroidTransform =
            Matrix.CreateTranslation(player.asteroidList[i].position);
        if (player.asteroidList[i].isActive)
        {
            DrawModel(asteroidModel, asteroidTransform,
                asteroidTransforms);
        }
    }

    for (int i = 0; i < GameConstants.NumBullets; i++)
    {
        if (player.bulletList[i].isActive)
        {
            Matrix bulletTransform =
                Matrix.CreateTranslation(player.bulletList[i].position);
            DrawModel(bulletModel, bulletTransform, bulletTransforms);
        }
    }
    spriteBatch.Begin(SpriteBlendMode.AlphaBlend,
        SpriteSortMode.Immediate, SaveStateMode.None);
}

```

```

        spriteBatch.DrawString(lucidaConsole, "Score: " + player.score,
                               scorePosition, Color.LightGreen);
        spriteBatch.End();
        base.Draw(gameTime);
    }

```

At this point, you should be able to compile and run your game! We have been making small, iterative changes to make the code more elegant, but when you run the game, it should still look essentially the same. In the next step we will begin making the changes to the game rendering to allow the players to see one another.

Step 3: Split the Screen

Now that we have a way to create multiple players, we need a place to display the second player. In Game1.cs, create three new member variables to store the main viewport, the left viewport, and the right viewport.

C#

```

Viewport mainWindowport;
Viewport leftViewport;
Viewport rightViewport;

```

When we use only half of the screen, the aspect ratio will change. In the constructor for the game, divide the aspect ratio by 2 to account for the split screen. If you compile and run your game after making this change, the game will be distorted due to the fact that the aspect ratio is no longer equivalent to the width and height of the back buffer.

C#

```

public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    // this game is split screen, so divide the aspect ratio by 2.
    aspectRatio = (float)GraphicsDeviceManager.DefaultBackBufferWidth /
        (2 * GraphicsDeviceManager.DefaultBackBufferHeight);
}

```

In **LoadContent**, initialize the values for each viewport. We first set the main viewport to equal the graphics device viewport. The left viewport and right viewport will have a width that is half of the main viewport, with the right viewport beginning one pixel past the center of the screen.

C#

```

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    shipModel = Content.Load<Model>("Models/p1_wedge");
    shipTransforms = SetupEffectDefaults(shipModel);
    asteroidModel = Content.Load<Model>("Models/asteroid1");
    asteroidTransforms = SetupEffectDefaults(asteroidModel);
    bulletModel = Content.Load<Model>("Models/pea_proj");
    bulletTransforms = SetupEffectDefaults(bulletModel);
    stars = Content.Load<Texture2D>("Textures/B1_stars");
    soundEngine = Content.Load<SoundEffect>("Audio/Waves/engine_2");
    soundEngineInstance = soundEngine.CreateInstance();
    soundHyperspaceActivation =
        Content.Load<SoundEffect>("Audio/Waves/hyperspace_activate");
    soundExplosion2 =
        Content.Load<SoundEffect>("Audio/Waves/explosion2");
    soundExplosion3 =
        Content.Load<SoundEffect>("Audio/Waves/explosion3");
    soundWeaponsFire =
        Content.Load<SoundEffect>("Audio/Waves/tx0_fire1");
    lucidaConsole = Content.Load<SpriteFont>("Fonts/Lucida Console");
}

```

```

// Initialize the values for each viewport
mainViewport = GraphicsDevice.Viewport;
leftViewport = mainViewport;
rightViewport = mainViewport;
leftViewport.Width = leftViewport.Width / 2;
rightViewport.Width = rightViewport.Width / 2;
rightViewport.X = leftViewport.Width + 1;
}

```

Notice that the **Draw** method in game contains all the code necessary to draw a player. Rename **Draw** to **DrawPlayer**, and change the method so that it returns nothing and accepts two arguments, the *Player* to draw, and the *Viewport* to draw the player in. At the beginning of **DrawPlayer**, set the graphics device viewport to the viewport argument. Don't forget to remove the call to the **Draw** method of the base class, located at the end of the function.

C#

```

void DrawPlayer(Player player, Viewport viewport)
{
    graphics.GraphicsDevice.Viewport = viewport;

    spriteBatch.Begin(SpriteBlendMode.None, SpriteSortMode.Immediate,
        SaveStateMode.None);
    spriteBatch.Draw(stars, new Rectangle(0, 0, 800, 600), Color.White);
    spriteBatch.End();

    Matrix shipTransformMatrix = player.ship.RotationMatrix
        * Matrix.CreateTranslation(player.ship.Position);
    if (player.ship.isActive)
    {
        DrawModel(shipModel, shipTransformMatrix, shipTransforms);
    }
    for (int i = 0; i < GameConstants.NumAsteroids; i++)
    {
        Matrix asteroidTransform =
            Matrix.CreateTranslation(player.asteroidList[i].position);
        if (player.asteroidList[i].isActive)
        {
            DrawModel(asteroidModel, asteroidTransform,
                asteroidTransforms);
        }
    }
    for (int i = 0; i < GameConstants.NumBullets; i++)
    {
        if (player.bulletList[i].isActive)
        {
            Matrix bulletTransform =
                Matrix.CreateTranslation(player.bulletList[i].position);
            DrawModel(bulletModel, bulletTransform, bulletTransforms);
        }
    }
    spriteBatch.Begin(SpriteBlendMode.AlphaBlend,
        SpriteSortMode.Immediate, SaveStateMode.None);
    spriteBatch.DrawString(lucidaConsole, "Score: " + player.score,
        scorePosition, Color.LightGreen);
    spriteBatch.End();
}

```

Finally, recreate the **Draw** method, calling **DrawPlayer** in this new, shorter method, and move the call to **base.Draw(gameTime)** into this method.

C#

```

protected override void Draw(GameTime gameTime)
{
    DrawPlayer(player, leftViewport);
    base.Draw(gameTime);
}

```

```
}
```

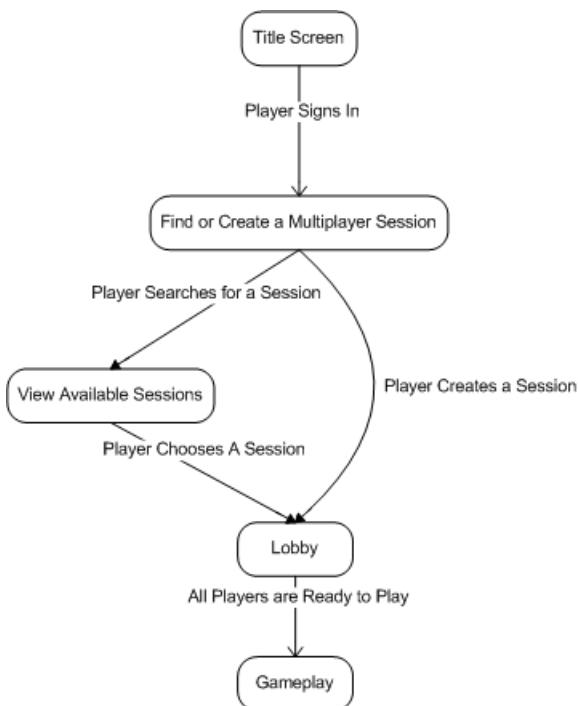
If you run your game now, you will see that the game for the local player is now showing on the left side of the screen:



Step 4: Game State Management

- [The Title Screen](#)
- [The Lobby](#)
- [List Available Network Sessions](#)

In a networked game, you will want the game to display different information depending on the state of the game. For example, if the player has not yet signed in, you will want to display a message with instructions on how to sign in. If the player would like to host or join an multiplayer game on the subnet, you will need to display a list of available games. Once a player selects a game to join you will need to allow the player to wait in the lobby until all players are ready to start the game.



To do this, we will create the main **Update** and **Draw** methods so that they choose the right area to update or draw - the title screen, the list of available sessions, the lobby, or the game. In this step of the tutorial, we will make separate update and draw methods to handle the particular requirements of the title screen, session selection screen, lobby, and game.

Begin by declaring the new global variables we will need in the game to manage the network session. Declare a variable to hold the current network session for the game, a collection of available network sessions, the index of the network session a player has selected to join, and a packet reader and writer to read network data.

C#

```
NetworkSession networkSession;  
AvailableNetworkSessionCollection availableSessions;
```



```
int selectedSessionIndex;
PacketReader packetReader = new PacketReader();
PacketWriter packetWriter = new PacketWriter();
```

While you are looking at the global variables for the game, comment out the **Player** instance that is declared there. We are going to instead associate the player data with a person who has signed into the session.

C#

```
//Player player = new Player();
```

To use network services in a game, we need to add a [GamerServicesComponent](#) to the game. Once this component has been added, we can respond to the event that occurs when a gamer signs in.

In the **Game** constructor, add a new [GamerServicesComponent](#) to the collection of game components. Also in the **Game** constructor, add a new event handler for the [SignedIn](#) event.

C#

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    aspectRatio = (float)GraphicsDeviceManager.DefaultBackBufferWidth /
        (2 * GraphicsDeviceManager.DefaultBackBufferHeight);

    // Add Gamer Services
    Components.Add(new GamerServicesComponent(this));

    // Respond to the SignedInGamer event
    SignedInGamer.SignedIn +=
        new EventHandler<SignedInEventArgs>(SignedInGamer_SignedIn);
}
```

One of the arguments that is passed to the [SignedIn](#) event is an instance of [SignedInEventArgs](#). This type contains a property called [Gamer](#). Each [Gamer](#) has a [Tag](#) object which can be used to attach data to the gamer. We are going to use the [Tag](#) property to store the **Player** that is associated with a particular gamer. Because this data does not exist yet when a gamer signs in, we will create a new **Player** when responding to the [SignedIn](#) event.

C#

```
void SignedInGamer_SignedIn(object sender, SignedInEventArgs e)
{
    e.Gamer.Tag = new Player();
}
```

We want to look at the input for the list of signed-in gamers during gameplay. Rename the current **Update** function **HandleGameplayInput** and set the accessibility level to "private." Also remove the call to **base.Update**. Change this method to accept a *Player* argument in addition to the *GameTime*. Also update the network session when this method is called.

C#

```
private void HandleGameplayInput(Player player, GameTime gameTime)
{
    if (IsButtonPressed(Buttons.Back))
        this.Exit();

    // change UpdateInput to take a Player
    UpdateInput(player);

    player.Update(gameTime);
    networkSession.Update();

    //base.Update(gameTime);
}
```

There is a static property called [SignedInGamers](#) that contains the list of all signed-in players. We will use this property in the **Update** method to get the **Player** data from a [SignedInGamer](#). Note that this loop wraps the update code that was already in this method. We also change **UpdateInput** to accept an instance of **Player**. This method also updates the **lastState** and **currentState** member variables, so the update of these variables can be removed from **UpdateInput**

C#

```
protected override void Update(GameTime gameTime)
{
    if (!Guide.IsVisible)
    {
        foreach (SignedInGamer signedInGamer in
            SignedInGamer.SignedInGamers)
        {
            Player player = signedInGamer.Tag as Player;
            lastState = player.lastState;
            currentState = GamePad.GetState(signedInGamer.PlayerIndex);

            if (networkSession != null)
            {
                // Handle the lobby input here...
            }
            else if (availableSessions != null)
            {
                // Handle the available sessions input here..
            }
            else
            {
                // Handle the title screen input here..
            }
            player.lastState = currentState;
        }
    }
    base.Update(gameTime);
}
```

The **UpdateInput** method will stay the same, except that it now gets the player to update from the *Player* argument instead of a global variable. There is also no need for the *currentState* and *lastState* variables to be updated in this method, so you can comment out these lines.

C#

```
protected void UpdateInput(Player player)
{
    //// Get the game pad state.
    //currentState = GamePad.GetState(PlayerIndex.One);
    //lastState = player.lastState;
    ...
    //player.lastState = currentState;
}
}
```

Finally, rename the **Draw** method to **DrawGameplay** and set the accessibility to "private." In this method, we will check to see if a network session has been created or joined before we start drawing the game. For each person in the networked game, we will draw to a different area of the screen.

C#

```
private void DrawGameplay(GameTime gameTime)
{
    GraphicsDevice.Viewport = mainViewport;
    GraphicsDevice.Clear(Color.CornflowerBlue);

    Player player;
    if (networkSession != null)
    {
```

```

        foreach (NetworkGamer networkGamer in networkSession.AllGamers)
        {
            player = networkGamer.Tag as Player;
            if (networkGamer.IsLocal)
            {
                DrawPlayer(player, leftViewport);
            }
            else
            {
                DrawPlayer(player, rightViewport);
            }
        }
    }
}

```

If you compile and run your game at this point, you will see a blank screen. Do not panic! This is because we made the game play state of the game contingent on the user signing in and creating or joining a network session. We will implement these game states next.

The Title Screen

First, create a method to draw the title screen.

C#

```

private void DrawTitleScreen()
{
    GraphicsDevice.Clear(Color.CornflowerBlue);
    string message = "";

    if (SignedInGamer.SignedInGamers.Count == 0)
    {
        message = "No profile signed in! \n" +
            "Press the Home key on the keyboard or \n" +
            "the Xbox Guide Button on the controller to sign in.";
    }
    else
    {
        message += "Press A to create a new session\n" +
            "X to search for sessions\nB to quit\n\n";
    }
    spriteBatch.Begin();
    spriteBatch.DrawString(lucidaConsole, message,
        new Vector2(101, 101), Color.Black);
    spriteBatch.DrawString(lucidaConsole, message,
        new Vector2(100, 100), Color.White);
    spriteBatch.End();
}

```

Next, change the **Draw** method so that it draws our new title screen if there is no network session available and no available sessions to list.

C#

```

protected override void Draw(GameTime gameTime)
{
    if (networkSession != null)
    {
    }
    else if (availableSessions != null)
    {
        // Show the available session...
    }
    else
    {
        DrawTitleScreen();
    }
}

```

```
    base.Draw(gameTime);  
}
```

We also need to handle the title screen input.

C#

```
protected void HandleTitleScreenInput()  
{  
    if (IsButtonPressed(Buttons.A))  
    {  
        CreateSession();  
    }  
    else if (IsButtonPressed(Buttons.X))  
    {  
        availableSessions = NetworkSession.Find(  
            NetworkSessionType.SystemLink, 1, null);  
  
        selectedSessionIndex = 0;  
    }  
    else if (IsButtonPressed(Buttons.B))  
    {  
        Exit();  
    }  
}
```

This method creates a session if the user selects this option.

C#

```
void CreateSession()  
{  
    networkSession = NetworkSession.Create(  
        NetworkSessionType.SystemLink,  
        1, 8, 2,  
        null);  
  
    networkSession.AllowHostMigration = true;  
    networkSession.AllowJoinInProgress = true;  
  
    HookSessionEvents();  
}
```

When creating a session, subscribe to the [GamerJoined](#) event.

C#

```
private void HookSessionEvents()  
{  
    networkSession.GamerJoined +=  
        new EventHandler<GamerJoinedEventArgs>(  
            networkSession_GamerJoined);  
}
```

When responding to the [GamerJoined](#) event, we want to either create a new **Player** if the player is not local, or get the player if the player has already signed in and has a **Player** object associated with it.

C#

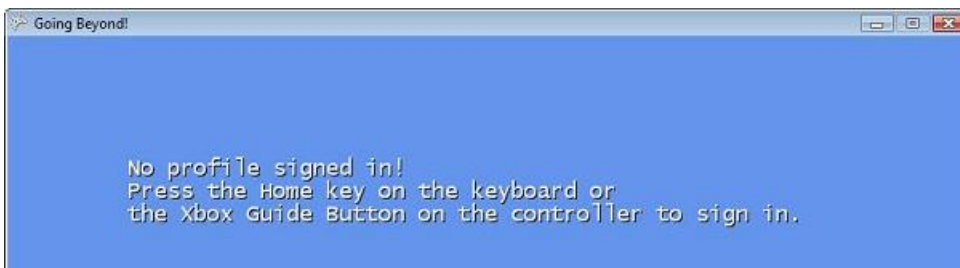
```
void networkSession_GamerJoined(object sender, GamerJoinedEventArgs e)  
{  
    if (!e.Gamer.IsLocal)  
    {  
        e.Gamer.Tag = new Player();  
    }  
    else  
    {  
        e.Gamer.Tag = GetPlayer(e.Gamer.Gamertag);  
    }  
}
```

```
}  
}
```

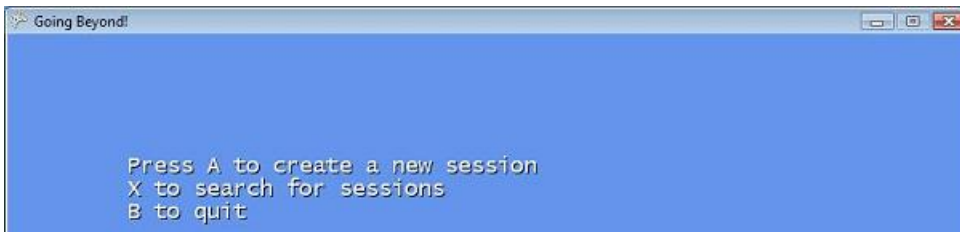
C#

```
Player GetPlayer(String gamertag)  
{  
    foreach (SignInGamer signInGamer in  
        SignInGamer.SignedInGamers)  
    {  
        if (signInGamer.Gamertag == gamertag)  
        {  
            return signInGamer.Tag as Player;  
        }  
    }  
  
    return new Player();  
}
```

If you run your game, you will see the new title screen with sign-in instructions for the player. Because we have a [GamerServicesComponent](#) added to the game, you can now sign in or view the Guide when you follow the instructions on the screen.



Once a player signs in, the instructions for finding or creating a new network session will also be displayed.



The Lobby

First, we will create a game screen to display the players waiting in the lobby.

C#

```
private void DrawLobby()  
{  
    GraphicsDevice.Clear(Color.CornflowerBlue);  
    spriteBatch.Begin();  
    float y = 100;  
  
    spriteBatch.DrawString(lucidaConsole, "Lobby (A=ready, B=leave)",  
        new Vector2(101, y + 1), Color.Black);  
    spriteBatch.DrawString(lucidaConsole, "Lobby (A=ready, B=leave)",  
        new Vector2(101, y), Color.White);  
  
    y += lucidaConsole.LineSpacing * 2;  
  
    foreach (NetworkGamer gamer in networkSession.AllGamers)  
    {  
        string text = gamer.Gamertag;  
  
        Player player = gamer.Tag as Player;  
  
        if (player.picture == null)  
        {
```

```

        GamerProfile gamerProfile = gamer.GetProfile();
        player.picture = gamerProfile.GamerPicture;
    }

    if (gamer.IsReady)
        text += " - ready!";

    spriteBatch.Draw(player.picture, new Vector2(100, y),
        Color.White);
    spriteBatch.DrawString(lucidaConsole, text, new Vector2(170, y),
        Color.White);

    y += lucidaConsole.LineSpacing + 64;
}
spriteBatch.End();
}

```

Next, we will need a method to handle any input from the user while the user is in the lobby.

C#

```

protected void HandleLobbyInput()
{
    // Signal I'm ready to play!
    if (IsButtonPressed(Buttons.A))
    {
        foreach (LocalNetworkGamer gamer in networkSession.LocalGamers)
            gamer.IsReady = true;
    }

    if (IsButtonPressed(Buttons.B))
    {
        networkSession = null;
        availableSessions = null;
    }

    // The host checks if everyone is ready, and moves
    // to game play if true.
    if (networkSession.IsHost)
    {
        if (networkSession.IsEveryoneReady)
            networkSession.StartGame();
    }

    // Pump the underlying session object.
    networkSession.Update();
}

```

Add code to the **Draw** method so that it will call the **DrawLobby** function if the user should be in the lobby.

C#

```

protected override void Draw(GameTime gameTime)
{
    if (networkSession != null)
    {
        //If the session is not null, we're either
        //in the lobby or playing the game...
        // Draw the Lobby
        if (networkSession.SessionState == NetworkSessionState.Lobby)
            DrawLobby();
    }
    else if (availableSessions != null)
    {
        // Show the available session...
    }
}

```

```

else
{
    DrawTitleScreen();
}

base.Draw(gameTime);
}

```

Finally, change the **Update** method of the game so it will call the method that handles the lobby input.

C#

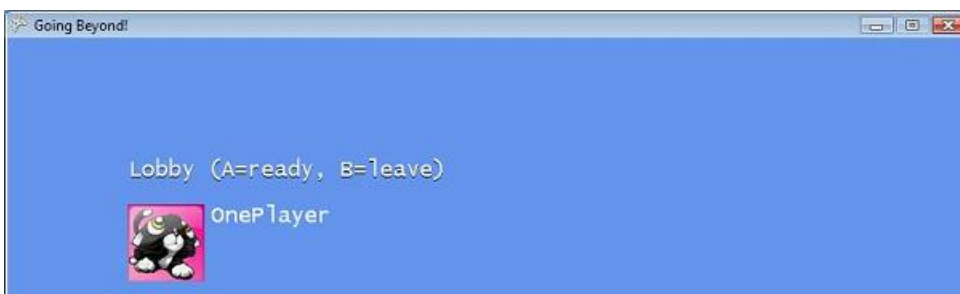
```

protected override void Update(GameTime gameTime)
{
    if (!Guide.IsVisible)
    {
        foreach (SignedInGamer signedInGamer in
            SignedInGamer.SignedInGamers)
        {
            Player player = signedInGamer.Tag as Player;
            lastState = player.lastState;
            currentState = GamePad.GetState(signedInGamer.PlayerIndex);

            if (networkSession != null)
            {
                if (networkSession.SessionState ==
                    NetworkSessionState.Lobby)
                    HandleLobbyInput();
            }
            else if (availableSessions != null)
            {
                // Handle the available sessions input here...
            }
            else
            {
                HandleTitleScreenInput();
            }
            player.lastState = currentState;
        }
    }
    base.Update(gameTime);
}

```

Compile and run the game. Sign in and create a session to see the lobby.



List Available Network Sessions

Just as we created a method to draw an update the other game states, we will also create methods to draw and update the list of available network sessions.

C#

```

private void DrawAvailableSessions()
{
    GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin();
    float y = 100;

    spriteBatch.DrawString(lucidaConsole,

```

```

        "Available sessions (A=join, B=back)",
        new Vector2(101, y + 1), Color.Black);
spriteBatch.DrawString(lucidaConsole,
    "Available sessions (A=join, B=back)",
    new Vector2(100, y), Color.White);

y += lucidaConsole.LineSpacing * 2;

int selectedIndex = 0;

for (
    int sessionIndex = 0;
    sessionIndex < availableSessions.Count;
    sessionIndex++)
{
    Color color = Color.Black;

    if (sessionIndex == selectedIndex)
        color = Color.Yellow;

    spriteBatch.DrawString(lucidaConsole,
        availableSessions[sessionIndex].HostGamertag,
        new Vector2(100, y), color);

    y += lucidaConsole.LineSpacing;
}
spriteBatch.End();
}

```

C#

```

protected void HandleAvailableSessionsInput()
{
    if (IsButtonPressed(Buttons.A))
    {
        // Join the selected session.
        if (availableSessions.Count > 0)
        {
            networkSession = NetworkSession.Join(
                availableSessions[selectedIndex]);
            HookSessionEvents();

            availableSessions.Dispose();
            availableSessions = null;
        }
    }
    else if (IsButtonPressed(Buttons.DPadUp))
    {
        // Select the previous session from the list.
        if (selectedIndex > 0)
            selectedIndex--;
    }
    else if (IsButtonPressed(Buttons.DPadDown))
    {
        // Select the next session from the list.
        if (selectedIndex < availableSessions.Count - 1)
            selectedIndex++;
    }
    else if (IsButtonPressed(Buttons.B))
    {
        // Go back to the title screen.
        availableSessions.Dispose();
        availableSessions = null;
    }
}
}

```


Change the **Draw** and **Update** methods to call these methods when the game should show the list of available sessions.

C#

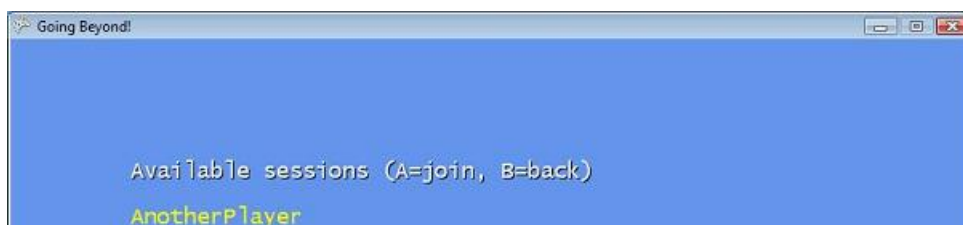
```
protected override void Draw(GameTime gameTime)
{
    if (networkSession != null)
    {
        //If the session is not null, we're either
        //in the lobby or playing the game...
        // Draw the Lobby
        if (networkSession.SessionState == NetworkSessionState.Lobby)
            DrawLobby();
    }
    else if (availableSessions != null)
    {
        DrawAvailableSessions();
    }
    else
    {
        DrawTitleScreen();
    }

    base.Draw(gameTime);
}
```

C#

```
protected override void Update(GameTime gameTime)
{
    if (!Guide.IsVisible)
    {
        foreach (SignedInGamer signedInGamer in
            SignedInGamer.SignedInGamers)
        {
            Player player = signedInGamer.Tag as Player;
            lastState = player.lastState;
            currentState = GamePad.GetState(signedInGamer.PlayerIndex);

            if (networkSession != null)
            {
                if (networkSession.SessionState ==
                    NetworkSessionState.Lobby)
                    HandleLobbyInput();
            }
            else if (availableSessions != null)
            {
                HandleAvailableSessionsInput();
            }
            else
            {
                HandleTitleScreenInput();
            }
            player.lastState = currentState;
        }
    }
    base.Update(gameTime);
}
```



When a player joins an available session, the other players waiting in the lobby can now be seen.



Step 5: Receive Network Data

We can now create a network session with multiple players in the session, but we need a way for players to send and receive data from other players in the game.

First, create a method to receive the data from a network gamer.

C#

```
void ReceiveNetworkData(LocalNetworkGamer gamer, GameTime gameTime)
{
    while (gamer.IsDataAvailable)
    {
        NetworkGamer sender;
        gamer.ReceiveData(packetReader, out sender);

        if (!sender.IsLocal)
        {
            Player player = sender.Tag as Player;
            player.ship.isActive = packetReader.ReadBoolean();
            player.ship.Position = packetReader.ReadVector3();
            player.ship.Rotation = packetReader.ReadSingle();
            player.score = packetReader.ReadInt32();
            if (packetReader.ReadBoolean())
            {
                player.ShootBullet();
            }
            if (packetReader.ReadBoolean())
            {
                player.ship.isActive = false;
            }
            for (int i = 0; i < GameConstants.NumAsteroids; i++)
            {
                player.asteroidList[i].isActive =
                    packetReader.ReadBoolean();
                player.asteroidList[i].position =
                    packetReader.ReadVector3();
            }
            player.Update(gameTime);
        }
    }
}
```

Next, change the **UpdateInput** method to call **ReceiveNetworkData** for every signed-in player on the local system that might receive data. Note that the **ReceiveNetworkData** needs an instance of **GameTime** to pass to the **Player.Update** method, so we will update **UpdateInput** to take the *GameTime* as a parameter. At the end of the **UpdateInput** Method, send the data from the local gamer to any network gamers in the game.

C#

```
private void HandleGameplayInput(Player player, GameTime gameTime)
{
    ...
    UpdateInput(player, gameTime);
    ...
}
```

```
}
```

C#

```
private void UpdateInput(Player player, GameTime gameTime)
{
    bool isFiring = false;
    bool shipDestroyed = false;

    foreach (LocalNetworkGamer gamer in networkSession.LocalGamers)
    {
        ReceiveNetworkData(gamer, gameTime);

        // this code is the same code we have been
        // using to update the player input
        if (currentState.IsConnected)
        {
            if (player.ship.isActive)
            {
                player.ship.Update(currentState);
                PlayEngineSound(currentState);
            }
            // In case you get lost, press B to warp back to the center.
            if (IsButtonPressed(Buttons.B))
            {
                player.WarpToCenter();
                // Make a sound when we warp.
                soundHyperspaceActivation.Play();
            }

            //are we shooting?
            if (player.ship.isActive && IsButtonPressed(Buttons.A))
            {
                player.ShootBullet();
                soundWeaponsFire.Play();
                isFiring = true;
            }

            if (player.CheckForBulletAsteroidCollision(
                bulletModel.Meshes[0].BoundingSphere.Radius,
                asteroidModel.Meshes[0].BoundingSphere.Radius))
            {
                soundExplosion2.Play();
            }

            shipDestroyed = player.CheckForShipAsteroidCollision(
                shipModel.Meshes[0].BoundingSphere.Radius,
                asteroidModel.Meshes[0].BoundingSphere.Radius);

            if (shipDestroyed)
            {
                soundExplosion3.Play();
            }
        }
        packetWriter.Write(player.ship.isActive);
        packetWriter.Write(player.ship.Position);
        packetWriter.Write(player.ship.Rotation);
        packetWriter.Write(player.score);
        packetWriter.Write(isFiring);
        packetWriter.Write(shipDestroyed);
        for (int i = 0; i < GameConstants.NumAsteroids; i++)
        {
            packetWriter.Write(player.asteroidList[i].isActive);
            packetWriter.Write(player.asteroidList[i].position);
        }

        gamer.SendData(packetWriter, SendDataOptions.None);
    }
}
```

```
}  
}
```

Finally, change **Draw** and **Update** so that each method calls the **DrawGameplay** and **HandleGameplayInput** method at the appropriate time.

C#

```
protected override void Draw(GameTime gameTime)  
{  
    if (networkSession != null)  
    {  
        // Draw the Lobby  
        if (networkSession.SessionState == NetworkSessionState.Lobby)  
            DrawLobby();  
        else  
            DrawGameplay(gameTime);  
    }  
    else if (availableSessions != null)  
    {  
        DrawAvailableSessions();  
    }  
    else  
    {  
        DrawTitleScreen();  
    }  
  
    base.Draw(gameTime);  
}
```

C#

```
protected override void Update(GameTime gameTime)  
{  
    if (!Guide.IsVisible)  
    {  
        foreach (SignedInGamer signedInGamer in  
            SignedInGamer.SignedInGamers)  
        {  
            Player player = signedInGamer.Tag as Player;  
            lastState = player.lastState;  
            currentState = GamePad.GetState(signedInGamer.PlayerIndex);  
  
            if (networkSession != null)  
            {  
                if (networkSession.SessionState ==  
                    NetworkSessionState.Lobby)  
                    HandleLobbyInput();  
                else  
                    HandleGameplayInput(player, gameTime);  
            }  
            else if (availableSessions != null)  
            {  
                HandleAvailableSessionsInput();  
            }  
            else  
            {  
                HandleTitleScreenInput();  
            }  
            player.lastState = currentState;  
        }  
    }  
    base.Update(gameTime);  
}
```

With this final change, your game can now be played by two players over the network.

Where do you go from here? Using the **Player** class, you could extend the game to allow for two players on the same local machine. You could also change the game to display more than two players by splitting the screen again, or get rid of the split-screen and implement code to allow the players to shoot at one another.

More advanced screen management techniques are available from the [XNA Creators Club Online Web site](#), along with numerous other game programming techniques that can be used to add functionality to this game.

Upgrading XNA Game Studio Projects

Guides for upgrading XNA Game Studio Projects are provided in the following sections:

In This Section

[Upgrade Guide: XNA Game Studio 3.0 to XNA Game Studio 3.1](#)

Describes how to upgrade your XNA Game Studio 3.0 game to XNA Game Studio 3.1 in Microsoft Visual Studio 2008.

[Upgrade Guide: XNA Game Studio 2.0 to XNA Game Studio 3.1](#)

Describes how to upgrade your XNA Game Studio 2.0 game to XNA Game Studio 3.1 in Microsoft Visual Studio 2008.

[Upgrade Guide: XNA Game Studio 2.0 to XNA Game Studio 3.0](#)

Describes how to upgrade your XNA Game Studio 2.0 game to target XNA Framework 3.0 in Microsoft Visual Studio 2008.

[Troubleshooting Upgrades](#)

Describes common issues with upgrading XNA Game Studio projects.

Choosing an XNA Framework Version

XNA Framework 3.0 and 3.1

XNA Game Studio 3.1 supports both XNA Game Studio 3.1 and XNA Game Studio 3.0 project formats. To target XNA Framework 3.1, use an XNA Game Studio 3.1 project. XNA Game Studio 3.0 projects will target XNA Framework 3.0.

Important

You should not mix projects that target different versions of the XNA Framework in the same solution.

Unless you have a special need to maintain an XNA Game Studio 3.0 project, you should upgrade your projects to XNA Game Studio 3.1 to take advantage of the new features in this release. For more information, see [What's New in XNA Game Studio 3.1](#).

XNA Game Studio 3.1 projects must use the Microsoft Cross Platform Audio Creation Tool version 3 (XACT3), which is new to XNA Game Studio 3.1. The version of XACT provided with XNA Game Studio 3.0, XACT2, writes project files (.xap) that are incompatible with XNA Game Studio 3.1 projects. XNA Game Studio 3.0 projects must continue to use XACT2. Both versions of XACT are available in the **Start** menu, under **All Programs | Microsoft XNA Game Studio 3.1 | Tools**.

XNA Framework 2.0

XNA Game Studio 2.0 projects, targeting XNA Framework 2.0, are not supported in this release. All XNA Game Studio 2.0 projects must be upgraded to target either XNA Framework 3.1 or XNA Framework 3.0.

For more information about upgrading XNA Game Studio 2.0 projects, see [Upgrade Guide: XNA Game Studio 2.0 to XNA Game Studio 3.0](#).

Upgrade Guide: XNA Game Studio 3.0 to XNA Game Studio 3.1

To take advantage of the latest XNA Game Studio features, projects should be upgraded to XNA Game Studio 3.1. However, unlike converting a project from XNA Game Studio 2.0 to XNA Game Studio 3.0, this operation is not enforced automatically.

For a quick understanding of the new features introduced in XNA Game Studio 3.1, see [What's New in XNA Game Studio 3.1](#).

Upgrading Your Projects

To upgrade projects from XNA Game Studio 3.0 to XNA Game Studio 3.1

1. Right-click your solution in Visual Studio's **Solution Explorer**.

This opens a context sensitive menu with the option: **Upgrade Solution...**

2. Click **Upgrade Solution...** to open the **Upgrade Solution** dialog box.

This lists all the projects in the current solution that will be upgraded.

3. If you are sure you want to upgrade now, click **Upgrade**.

This upgrades all the listed projects to XNA Game Studio 3.1. You will be presented with a dialog box that prompts you to save your existing projects.

Click **Yes** to save your projects and continue with the upgrade. If you click **No** or **Cancel**, your project will not be upgraded. You *must* save the files before you upgrade them.

Important

This step will only save your project files and continue with the upgrade. It does not back up your projects. Be sure to back up your project files *before* continuing with the upgrade. You cannot Undo the upgrade once it begins.

Note

If you have any Microsoft Cross-Platform Audio Creation Tool (XACT) project files (.xap) that were created with XNA Game Studio 3.0, they must be loaded and saved with XACT3 or they will not work with XNA Game Studio 3.1. XACT3 can be launched from the **Start** menu, by clicking **All Programs, Microsoft XNA Game Studio 3.1, Tools, and Microsoft Cross-Platform Audio Creation Tool 3 (XACT3)**. For more information about XACT, see [Audio Overview](#).

The chief tasks that the Project Upgrade Wizard performs are:

- Updates XNA Framework references and properties from XNA Game Studio 3.0 to XNA Game Studio 3.1.
- Updates the import targets file references in the game project for what is required for XNA Game Studio 3.1.
- Updates the import targets file references in the content project for what is required for XNA Game Studio 3.1.

Upgrade Guide: XNA Game Studio 2.0 to XNA Game Studio 3.1

Describes how to upgrade your XNA Game Studio 2.0 game to XNA Game Studio 3.1 in Microsoft Visual Studio 2008.

XNA Game Studio 2.0 game projects require upgrade to the new format for development in Visual Studio 2008 and XNA Game Studio 3.1. This upgrade is necessary in every case, regardless of the complexity or simplicity of the project.

For a quick understanding of the new features introduced in XNA Game Studio 3.1, see [What's New in XNA Game Studio 3.1](#).

Upgrading Your Projects

Projects that were created in XNA Game Studio 2.0 must be upgraded to work with Microsoft Visual Studio 2008 and *at least* XNA Game Studio 3.0. There is no direct way to upgrade a project from XNA Game Studio 2.0 to XNA Game Studio 3.1. However, to upgrade a project to XNA Game Studio 3.1, it must first be upgraded to XNA Game Studio 3.0.

To upgrade a project from XNA Game Studio 2.0 to XNA Game Studio 3.1

1. Upgrade the project from XNA Game Studio 2.0 to XNA Game Studio 3.0, using the procedure in [Upgrade Guide: XNA Game Studio 2.0 to XNA Game Studio 3.0](#).
2. Upgrade the project from XNA Game Studio 3.0 to XNA Game Studio 3.1, using the procedure in [Upgrade Guide: XNA Game Studio 3.0 to XNA Game Studio 3.1](#).

Upgrade Guide: XNA Game Studio 2.0 to XNA Game Studio 3.0

Describes how to upgrade your XNA Game Studio 2.0 game to target XNA Framework 3.0 in Microsoft Visual Studio 2008.

XNA Game Studio 2.0 game projects must be upgraded to target either XNA Framework 3.1 or XNA Framework 3.0 for development in Visual Studio 2008 and XNA Game Studio 3.1. This upgrade is necessary in every case, regardless of the complexity or simplicity of the project.

Note

To take advantage of the latest features in XNA Game Studio, you should upgrade your projects to XNA Game Studio 3.1. To do so, first follow the procedure in this guide to upgrade the project to XNA Game Studio 3.0. When that is complete, see [Upgrade Guide: XNA Game Studio 3.0 to XNA Game Studio 3.1](#). XNA Framework 3.0 games are still supported in XNA Game Studio 3.1, however.

For a quick understanding of the features introduced in XNA Game Studio 3.0, see [XNA Game Studio 3.0: What's New in This Release](#).

Upgrading Your Projects

Projects created in XNA Game Studio 2.0 must be upgraded to work with Microsoft Visual Studio 2008 and XNA Game Studio 3.0.

The Project Upgrade Wizard for XNA Game Studio 3.0 executes automatically whenever you open a project or solution in Visual Studio 2008 that was created for XNA Game Studio 2.0. The instructions provided by the Project Upgrade Wizard will lead you through the upgrade process.

The chief tasks that the Project Upgrade Wizard performs are:

- Updates XNA Framework references and properties from XNA Framework 2.0 to XNA Framework 3.0.
- Updates the import targets file references in the game project for what is required for XNA Framework 3.0.
- Updates the import targets file references in the content project for what is required for XNA Framework 3.0.
- Establishes content compression properties (introduced in XNA Game Studio 3.0) to the default settings.

Troubleshooting Upgrades

Projects don't build because a reference to assembly version X is missing, but I am not targeting X.

Be sure to check the dependencies of all assemblies included with your projects. Also, make sure that any assemblies you're using in your projects target the same version of the XNA Framework that your project is targeting. Projects (or libraries) that target different versions of the XNA Framework cannot interact and should not be located in the same solution.

Projects don't build because a type can't be converted from type Foo (assembly version X) to type Foo (assembly version Y).

Check to see that all classes in your project use the classes, methods, properties, and events that exist in the XNA Framework version you are targeting. In some cases, methods may have been deprecated or some parameters may have changed.

My importer or processor is missing from the list of importers or processors available in the content pipeline.

The importer or processor used by your project may not support the version of XNA Framework that you are targeting. Be sure that you have updated versions of all components used by your title.

My content won't build because the importer or processor can't be found, but I have a reference to it.

The importer or processor used by your project may not support the version of XNA Framework that you are targeting. Be sure that you have updated versions of all components used by your title.

My audio project files are not loaded in my converted solution.

Audio project files (.xap) that were created with XACT2 in XNA Game Studio 3.0 must be loaded and saved with XACT3 before they will work with XNA Game Studio 3.1. For more information, see [Upgrade Guide: XNA Game Studio 3.0 to XNA Game Studio 3.1](#).

See Also [Upgrading XNA Game Studio Projects](#)

Using XNA Framework Starter Kits

An XNA Framework Starter Kit is a complete or near-complete, self-contained game that includes both game code and game assets. A starter kit is ready for you to load and build. Each kit comes with its own documentation, including descriptions of programming techniques, and suggestions for how it may be customized. A starter kit provides a great way to see a working XNA Framework game in action and to see what is possible using XNA Game Studio for game development.

To Load and Build an XNA Framework Starter Kit

1. From the **File** menu, click **New**, and then click **New Project**.

The **New Project** dialog box appears.

2. In **Project types**, under the **Visual C#** node, select **XNA Game Studio 3.1**.

The **New Project** dialog box displays the XNA Framework–specific project types.

Next, you need to select an XNA Framework Starter Kit project type.

3. Select an available starter kit and then click **OK**.

The selected starter kit loads into your version of Visual Studio.

4. To build and launch the starter kit project, press F5.

Starter Kit: Platformer

The Platformer Starter Kit is a near-complete, self-contained game solution that includes both game code and game assets. The game is a standard 2D platformer with levels, enemies, and collectable gems.

This starter kit is intentionally incomplete. Several nonessential features and systems are not finished. This makes it easier and quicker to understand the structure of the game, and how the pieces fit together to provide a rich gaming experience. The Platformer Starter Kit includes the following features:

- Cross-platform support for Windows, Zune, and Xbox 360.
- Control of the player character using either the keyboard or gamepad.
- Simple physics modeling (falling and jumping) and dynamic collision checking.
- Production-level sprite sheets, sound effects, and other game assets.
- High and low resolution assets, and an additional content project containing audio assets.
- New features from XNA Game Studio, such as the simple sound API and multiple content projects whose usage depends on the target platform.



Figure 1. Gameplay Screen for the Platformer Starter Kit

Player Controls

The player character is controlled using either the keyboard or gamepad.

Action	Keyboard	Gamepad	Zune
Run left, Run right	A, D	Left thumbstick or analog D-pad	Device control pad (version 1) or Zune pad (version 2)
Jump	Space	A	Click the center of either the device control pad (version 1) or the Zune pad (version 2)

Platformer Code Architecture

The following is a list of classes shipped with the Platformer starter kit. The file containing the implementation for each class shares the name of the class. For example, Gem.cs contains the Gem class implementation.

Name	Description
Gem	Implements a floating gem in the game. Gems are collectable by the player, and are worth a set amount of points. Gems are used to load, draw, and update a gem. For more information, see Basic Platformer Features .
Circle (structure)	Implements a bounding circle for checking collision against gem objects. For more information, see Basic Platformer Features .

Tile (structure)	Stores basic information about a game tile. For more information, see Basic Platformer Features .
TileCollision (enumeration)	Stores the different collision types a tile can have: Passable, Impassable, Platform. For more information, see Basic Platformer Features .
AnimationPlayer (structure)	Implements playback of the animation stored by <code>Animation</code> . For more information, see Intermediate Platformer Features .
Animation	Stores an animated texture. Used to animate the player character and enemy sprite sheets. For more information, see Intermediate Platformer Features .
Enemy	Implements an enemy in the game. Used to load, draw, and update an enemy. For more information, see Intermediate Platformer Features .
FaceDirection (enumeration)	Stores the different directions an enemy can face. For more information, see Intermediate Platformer Features .
PlatformerGame	Implements major game components such as content and level loading, HUD management and display, and game object updating. For more information, see Advanced Platformer Features .
Level	Implements a level in the game. A <code>Level</code> object contains a multi-dimensional array of tiles, a player character, a start and end location, point total, remaining level time, a list of gems, and a list of enemies. For more information, see Advanced Platformer Features .
Player	Implements the player character. Used to load, draw, and update the character. For more information, see Advanced Platformer Features .
RectangleExtensions	Implements an extension to the standard XNA Framework structure Rectangle . For more information, see Advanced Platformer Features .

Execution Flow

Think of the execution flow of Platformer as follows:

1. The next level is loaded. Important methods are `PlatformerGame.LoadContent` and `PlatformerGame.LoadNextLevel`.
2. The game is updated using `PlatformerGame.Update` and `Level.Update`. If the player is dead or if time has expired, input is ignored and the game is in a pause state. If the player has reached the exit, the remaining time is converted to points. If there is still time and the player hasn't reached the exit location, the time remaining is decremented and all level objects are updated (player character, enemies, gems, and so on) using their `Update` methods. At this time, checks are also made for the player reaching the exit and falling off the edge of the screen.
3. The gameplay screen is drawn using `PlatformerGame.Draw`. This method, in turn, calls `Level.Draw` and `PlatformerGame.DrawHud`.

`Level.Draw` is responsible for drawing the tiles, player character, gems and enemies using calls to the `Draw` method of each previously-mentioned game object.

Exploring the Platformer Starter Kit

[Basic Platformer Features](#)

Discusses basic features of the Platformer Starter Kit and offers recommendations for making basic modifications to the Platformer game.

[Intermediate Platformer Features](#)

Discusses intermediate classes of the Platformer Starter Kit, and offers recommendations for modifying or extending Platformer features.

[Advanced Platformer Features](#)

Discusses advanced features of the Platformer Starter Kit, and offers recommendations for modifying or extending Platformer features.

[Platformer: Adding Touch Support](#)

Extends the base Platformer starter kit code by adding touch screen and accelerometer support for input.

See Also

Concepts

[Using XNA Framework Starter Kits](#)

Tasks

[Your First Game: Microsoft XNA Game Studio in 2D](#)

[Platformer: Adding a Power-Up](#)

[Platformer: Adding a Scrolling Level](#)

Basic Platformer Features

Discusses basic features of the Platformer Starter Kit and offers recommendations for making basic modifications to the Platformer game.

- [Platformer Game Design](#)
- [Starter Kit Assets](#)
- [Exploring the Basic Classes](#)
- [Basic Modifications to the Platformer Starter Kit](#)

Platformer Game Design

The design of Platformer should be familiar to all gamers. You must avoid the enemies and reach the level exit before time expires (collecting gems along the way). If you complete the level with time remaining, that time is converted to a point bonus, and it is added to your current score. You lose a life if you run into an enemy, fall off a ledge, or run out of time before collecting all gems and reaching the exit.



Figure 1. Gameplay Screen for the Platformer Starter Kit

Game World

The game world is composed of individual levels where each level is automatically constructed, using several square tile types, from an existing text file. This text file uses a set of symbols to map out the level, locating the start and exit points, enemy starting positions, gem locations and tile locations (including their type). For a complete listing of the level file format, see [Advanced Platformer Features](#).

Three tile types are implemented by Platformer and represent the ledges and impassable areas of a level.

N Properties	
a m e	
I m p a s s a b l e	Players and enemies cannot pass through the tile from any direction.
P a s s a b l e	Players and enemies can pass through the tile freely. The main purpose of this tile type is to provide decoration to the level – decoration such as jungle foliage, rocks, and other scenery.

Platformer	Platform tiles behave like passable tiles except that players and enemies can stand on (or in the case of the player character, fall onto) the top edge of the tile without falling.
	This is implemented by checking for collision between the character's bounding rectangle and the top edge of the platform tile. If collision occurs, the affected character's Y velocity (that is, the character's speed when falling) is forced to 0. If the character stood, or fell onto, a passable tile, no collision checking is performed. This causes the character to continue falling until the character reaches an impassable tile or passes the bottom edge of the screen. For more information on collision checking and game world physics, see Advanced Platformer Features .

In addition to tiles, there are several special objects that represent special locations or objects in the game world.

Name	Properties
Level Start	The player character begins the level at this location, facing to the right.
Gem	An object that players collect for points and level completion. The player must collect <i>all</i> gems and reach the exit before time expires to complete the level.
Level Exit	The location the player must reach to complete the level before time expires. Failure to reach the exit (and collect all gems) forces the player to lose a life and begin at the Level Start location.

Player Character

The player character can run and jump. The player is affected by a simple gravity implementation, and can run off platforms without jumping. When falling, the character is halted by any collision with a platform-type tile. If the player character is moving and not under direct control of the player, simple pseudo-drag is implemented. This gradually stops the player character instead of causing an instant stop.

Enemies

Initially, Platformer implements a single enemy type. This enemy slowly walks back and forth along a platform, automatically turning around at the platform's edge. The player character cannot kill the enemy and is immediately killed by the enemy upon contact.

Starter Kit Assets

Assets in the various content projects are platform-dependent. For Windows and Xbox 360 projects, the HighResolutionContent content project is used. It contains fonts, level backgrounds, higher-resolution tile textures (64x48) and single-animation sprite sheets. Each sprite list is a strip of single frames (96x96) that provide animation when displayed sequentially. For instance, the Celebrate.png asset is a series of frames of a celebration by the player character. The entire set of animation lists for a specific character (either enemy or player) are kept in a separate directory.

Zune projects use a lower resolution set of these level backgrounds, tile textures, fonts, and single-frame sprite lists. For example, tile texture resolutions are 32x32 and sprite lists contain frames that are half the size of their higher-resolution counterpart, 48x48.

In addition to the HighResolutionContent and LowResolutionContent content projects, all projects use an additional content project, SharedContent. This content project contains the sound effects for the game. It includes sound effects used when the player character collects a gem, reaches the level exit, or dies by falling or enemy contact. Unlike texture files, sound files are shared between all three platforms. Having a separate set of lower fidelity sound effects for the Zune would not be worth the effort and loss in quality when compared to the size saved when Platformer is installed.

Shared projects (such as SharedContent) are the default in XNA Game Studio, but it is also possible to set up platform-specific game content projects. It's easier to implement if you do a little planning before adding new projects to your solution. For more information on sharing game assets between projects, see [Platform-Specific Content Projects](#).

Exploring the Basic Classes

The Platformer starter kit can be pretty overwhelming the first time you see it. However, it can be broken down into more manageable pieces, based on the complexity of the game object and its usage by the Platformer game. In this section, we'll explore Platformer's basic classes and structures. We'll talk about their purpose and design and point out features that can be modified or expanded on. For information on more complex classes, see [Intermediate Platformer Features](#) and [Advanced Platformer Features](#).

Gem Class and Circle Structure

The `Gem` class implementation is located in `Gem.cs`. Important methods include:

LoadContent

Loads a grayscale gem-shaped texture.

Update

Updates the gem height at regular intervals during gameplay. This method is used to cause all gems in the level to oscillate up and down in relation to each other. The motion is implemented with a sine curve over time. In addition, the X-coordinate of the gem is used to produce a nice syncopated pattern.

Draw

Draws the gem using the specified color shade. The default color is yellow.

OnCollected

Plays a sound indicating retrieval by the player. This is an great place to modify the default behavior of gem collection!

Properties include the parent `Level`, its position in world space, and a bounding circle.

The bounding circle (implemented by the `Circle` structure, located in `Circle.cs`) is used to determine if the player is colliding with the gem. If the player collides with the gem, it is removed from the gem collection of the parent `Level` and points are added to the player's score. The most important method of `Circle` is `Intersects`. This method checks for intersection with a rectangle (used to represent the bounding area of other game objects). This method is called by `Level.UpdateGems` and if it returns **true**, the gem is removed and the `OnGemCollected` method is called. For more information on `OnX` methods, see [Changing the Behavior of Existing Game Events](#).

Gems are the only objects that use a bounding circle. All other objects use a bounding rectangle.

Gem Animation

The default animation of gems is based on a sine wave. They slowly rise and fall as a group during gameplay. The `Gem.Update` method controls this animation.

Tip

You can easily modify this behavior by changing the default height and speed of the animation.

Tile Structure and TileCollision Enumeration

The `Tile` structure implementation is located in `Tile.cs`. It has a constructor, but its main purpose is storing the properties of a level tile.

Properties include the tile dimensions (width and height), the collision behavior of the tile object, and the texture used when drawing the tile object.

The collision behavior (stored in the `Collision` property) is the most important property of the `Tile` structure. This property determines what kind of collision detection is done, if any. For more information on supported tile types, see [Game World](#).

The `TileCollision` enumeration lists all possible collision behaviors for a tile and is located in `Tile.cs`.

Basic Modifications to the Platformer Starter Kit

Changing the Behavior of Existing Game Events

Tip

Modifying default behaviors in the game is as easy as modifying the related `OnXEvent` method.

Platformer handles important events in the game using methods that follow an `OnX` naming convention, where X is an important event name. For instance, when the player character collects a gem, `OnCollected` is ultimately called. The complete list of methods called before `Gem.OnCollected` is as follows:

1. `Level.Update`
2. `Level.UpdateGems`
3. `Level.OnCollected`
4. `Gem.OnCollected`

When a gem is collected, the default behavior is to play a "gem collected" sound (`GemCollected.wma`). If you wanted to change this behavior, there are two places you should focus on: the `Level.OnCollected` and `Gem.OnCollected` methods. Add (or remove existing) code to do different things. For instance, to create a "cursed" gem (subtracts points from the player's total when collected), modify the `Level.OnCollected` method to subtract `Gem.PointValue` from the total instead of adding to it.

See Also

Concepts

[Starter Kit: Platformer](#)

Intermediate Platformer Features

Discusses intermediate classes of the Platformer Starter Kit, and offers recommendations for modifying or extending Platformer features.

In this section, we'll explore Platformer's intermediate features and the classes that implement them, such as sprite animation. We'll talk about their purpose and design, and point out features that can be modified or expanded on. For information on basic classes, see [Exploring the Basic Classes](#). For advanced classes, see [Advanced Platformer Features](#).



Figure 1. Jumping Animation for Player Character

Exploring the Intermediate Classes

The following classes are more complex than the basic Platformer classes, and they implement some intermediate features of Platformer.

AnimationPlayer Structure and Animation Class

The `AnimationPlayer` structure is used to animate the player character and the enemies of the current level. It stores a single animation sprite sheet in the `Animation` property, of type `Animation`. Animation begins with a call to the `PlayAnimation` method, but if the animation is already playing, the method immediately returns. This prevents the animation from being interrupted for any reason.

The animation infrastructure is intentionally simple and customized for the base version of Platformer. It is not designed to scale up or be used independently of Platformer. The only requirement is that the animation frame is square in shape.

Platformer provides five animations for the player character and three animations for each of the different enemies. However, not all are initially used in the game.

Note
For enemies, the Die animation is not used.

The `AnimationPlayer` structure implementation is located in `AnimationPlayer.cs`. Important methods include:

- `Draw`
Advances the current time of the animation and, based on the time, draws a single frame from the related sprite sheet.
- `PlayAnimation`
Begins playing a specified animation, using a single-animation sprite sheet. If the specified animation is already running, it is not interrupted.

Properties include the sprite sheet used for animation, the current frame index, and the origin of the current frame.

The `Animation` class implementation is located in `Animation.cs`. It stores important attributes of an animation, such as frame count, texture used for animation, frame width, and so on.

Enemy Class and FaceDirection Enumeration

The `Enemy` class implementation is located in `Enemy.cs`. Important methods include:

LoadContent

Loads a specific enemy sprite strip. The enemy type is determined by the level structure file. Platformer implements four different enemy types. These types differ only in appearance. They are stored in separate directories in the appropriate content project, under the Sprites directory. Each enemy type has two animations: Run and Idle. The third animation, Die, is not used, but it is included for future development.

Update

Updates the enemy location. The default enemy behavior is to pace back and forth across a platform, pausing at the platform's edge.

If an enemy detects that it will move into an impassable tile (such as a wall) or walk off the edge of a platform tile, the value of the `FaceDirection` property is reversed, causing the enemy to begin walking in the opposite direction.

Draw

Draws the enemy character using the specified animation frame.

Before drawing the current animation, `Draw` checks the current orientation of the enemy. If the enemy is facing to the right, the current frame is flipped by specifying `SpriteEffects.FlipHorizontally` in the call to `SpriteBatch.Draw`.

Properties are similar to the `Gem` class: the parent `Level`, its position in world space, and a bounding rectangle. This differs from the gem object implementation, which used a circle. A rectangular bounding rectangle makes more sense for enemy characters (and the player character) because of their interaction with the rectangular tiles that make up the level – especially the platform tiles.

The `Enemy` class is a simpler version of the `Player` class because it cannot jump and has only a single behavior: relentless pacing upon its current platform.

Expanding Platformer

A logical expansion for Platformer is the use of power-ups in the game. In this topic, an Invincibility power-up is implemented in the default version of the Platformer starter kit. This power-up gives the player character temporary invincibility against enemies is indicated by a red gem. For complete details on this expansion, see [Platformer: Adding a Power-Up](#).

See Also

Concepts

[Starter Kit: Platformer](#)

Tasks

[Platformer: Adding a Scrolling Level](#)

Platformer: Adding a Power-Up

Extends the base Platformer starter kit code by adding a power-up gem and player character effect.

Tip

It is *highly* recommended that you are already familiar with the structure and features of the Platformer starter kit. This extension involves modifications to several files in the Platformer starter kit. For more information on the Platformer starter kit, see [Starter Kit: Platformer](#).

This extension modifies five areas of the Platformer starter kit. It is recommended that you use the base Platformer starter kit solution as the starting point for your modifications.

Adding a power-up gem involves the following major steps:

- Modifying Gem.cs to support a power-up attribute and draw the power-up gem in a special way.
- Modifying AnimationPlayer.cs to draw a special effect on the player when invincible.
- Modifying Player.cs to support and monitor a power-up state, drawing a special effect when the character is powered-up.
- Modifying Enemy.cs to support being killed by a powered-up player character and to draw a Die animation.
- Modifying Level.cs to recognize and load a power-up gem from a level structure file. The level code must also notify other game objects that a power-up gem has been collected.



Figure 1. Power-Up Gem Extension

Modifying the Gem Class

Modification of the `Gem` class begins by changing some existing properties and adding a `IsPowerUp` property. Modify the existing code for the `PointValue` and `Color` properties to match the following:

C#

```
public readonly int PointValue;
public bool IsPowerUp { get; private set; }
public readonly Color Color;
```

This modification also adds the new property, `IsPowerUp`. Modify the existing `Gem` constructor to accept a new `isPowerUp` parameter. After the modification, it should match the following:

C#

```
public Gem(Level level, Vector2 position, bool isPowerUp)
{
```

```
    ...  
}
```

In the existing constructor, after the code setting the level and position of the gem object, add the following:

C#

```
    IsPowerUp = isPowerUp;  
    if (IsPowerUp)  
    {  
        PointValue = 100;  
        Color = Color.Red;  
    }  
    else  
    {  
        PointValue = 30;  
        Color = Color.Yellow;  
    }
```

This code sets the `IsPowerUp` property of the gem, and then raises the point value if it is a power-up.

The last modification to the `Gem` class is to call the `PowerUp` method (a method added later) of the proper player character. Add the following code after any existing code in the `OnCollected` method:

C#

```
    if (IsPowerUp)  
        collectedBy.PowerUp();
```

This completes the modification of `Gem.cs`. Let's move on to `AnimationPlayer.cs`, and add support for drawing the power-up effect.

Modifying the AnimationPlayer Class

Your modifications in this class are small. You'll add code using the new color being passed in to tint the player character sprite, and you'll add a new `Draw` method.

Modify the existing `Draw` method declaration to accept an additional parameter called `color`. This indicates the current color to use as a tint when drawing the invincible player character. After the modification, it should look like the following:

C#

```
    public void Draw(GameTime gameTime, SpriteBatch spriteBatch, Vector2 position, SpriteEffects spriteEffects, Color color)  
    {  
        ...  
    }
```

Now that we have a tint color being passed in, let's use it in our `Draw` call. Staying within this method, modify the final call to `Draw` to match the following:

C#

```
        spriteBatch.Draw(Animation.Texture, position, source, color, 0.0f, Origin, 1.0f, spriteEffects, 0.0f);
```

Now add a new `Draw` method, after the previous one, to be called when the power-up effect is not needed:

C#

```

public void Draw(GameTime gameTime, SpriteBatch spriteBatch, Vector2 position, SpriteEffects spriteEffects)
{
    Draw(gameTime, spriteBatch, position, spriteEffects, Color.White);
}

```

That was the last of the code for the strobing tint effect. Let's move on to the `Player` class, found in the `Player.cs` file, and add that `PowerUp` method, among other modifications.

Modifying the Player Class

Modification of the `Player` class begins by adding support for a power-up state. After the `IsAlive` property, add the following code:

C#

```

// Powerup state
private const float MaxPowerUpTime = 6.0f;
private float powerUpTime;
public bool IsPoweredUp
{
    get { return powerUpTime > 0.0f; }
}
private readonly Color[] poweredUpColors = {
    Color.Red,
    Color.Blue,
    Color.Orange,
    Color.Yellow,
};

private SoundEffect powerUpSound;

```

This code adds some properties to monitor the power-up time, an `IsPoweredUp` method, a `Color` array that stores the colors used to produce the special tinting of the player character, and a sound effect that plays while the character is powered up.

While you're here, add the `PowerUp` sound effect to the **SharedContent** content project.

1. Right-click the **Sounds** directory of the **SharedContent** content project, and select **Add**, then **Existing Item...**
2. Select the `PowerUp.wma` file, and then click **OK**.
3. Select the new sound effect, and change the **Content Processor** to **SoundEffect** (using the **Properties** window).

Load the new sound effect by adding the following line to the `LoadContent` method:

C#

```
powerUpSound = Level.Content.Load<SoundEffect>("Sounds/PowerUp");
```

Modify the `Reset` method to set the `powerUpTime` property to 0.

C#

```
powerUpTime = 0.0f;
```

In the `Update` method, you'll need to check to see if the player character is powered up and, if so, update the time remaining. Add the following code after the `ApplyPhysics` call:

C#


```
    if (IsPoweredUp)
        powerUpTime = Math.Max(0.0f, powerUpTime - (float)gameTime.ElapsedGameTime.TotalSeconds);
```

We'll use the `Draw` method to indicate that the player is temporarily invincible. A strobing tint is applied while the player is invincible. Add the following code after the code that flips the player character sprite:

C#

```
    // Calculate a tint color based on power up state.
    Color color;
    if (IsPoweredUp)
    {
        float t = ((float)gameTime.TotalGameTime.TotalSeconds + powerUpTime / MaxPowerUpTime)
* 20.0f;
        int colorIndex = (int)t % poweredUpColors.Length;
        color = poweredUpColors[colorIndex];
    }
    else
    {
        color = Color.White;
    }
```

If the player is invincible, a color is selected from the array you added earlier and used to tint the character sprite when it is later drawn. The final modification to this function is to add the tint color to the call to `AnimationPlayer.Draw`. Modify the call to match the following:

C#

```
sprite.Draw(gameTime, spriteBatch, Position, flip, color);
```

The final bit of code is the new `PowerUp` method. Add this new code after the `Draw` method:

C#

```
public void PowerUp()
{
    powerUpTime = MaxPowerUpTime;
    powerUpSound.Play();
}
```

This is a simple bit of code that sets the powerup time to a predefined value, which is 3 seconds in this case.

You're done with `Player.cs`. It's now time to modify his fiendish enemies.

Modifying the Enemy Class

Modification of the `Enemy` class begins by adding and loading the death animation. You'll need to add the `Die` animation (`Die.png`) to both the `HighResolutionContent` and `LowResolutionContent` content projects for each monster type.

1. From **Solution Explorer**, expand the **HighResolutionContent** icon, and navigate down to the `\Sprites\MonsterA` directory.
2. Right-click the **MonsterA** icon, and use the **Add Existing Item** dialog box to add the `Die.png` game asset.
3. Do the same for the `MonsterA` directory located in the **LowResolutionContent** content project.
4. Repeat Steps 1-3 for each remaining monster type.

Add a declaration for a new `Enemy` property of type `Animation` to the existing animation list, called `dieAnimation`.

C#

```
private Animation dieAnimation;
```

You'll want a sound effect to play when an enemy dies, so go ahead and add the existing `MonsterKilled.wma` game asset to the `Sounds` directory of the **SharedContent** content project using the **Add Existing Item** dialog box. Now you need a new variable for storing that sound effect. Add the following code after the animation variables:

C#

```
// Sounds  
private SoundEffect killedSound;
```

Add another property after the existing `BoundingBox` variable:

C#

```
public bool IsAlive { get; private set; }
```

Now that an enemy can be killed by the player character, you need to track whether enemies are alive or dead. In the `Enemy` constructor (right below the last modification), initialize `IsAlive` to **true**.

C#

```
this.IsAlive = true;
```

In the `LoadContent` method, load the death animation:

C#

```
dieAnimation = new Animation(Level.Content.Load<Texture2D>(spriteSet + "Die"), 0.07f, false);
```

And then the sound effect:

C#

```
// Load sounds.  
killedSound = Level.Content.Load<SoundEffect>("Sounds/MonsterKilled");
```

Since this is the first sound effect for enemies, you'll need to add a reference to `Microsoft.Xna.Framework.Audio` to this file. Add the following code after any existing **using** code:

C#

```
using Microsoft.Xna.Framework.Audio;
```

Now that enemies can be dead, optimize the update code a bit, and only update living enemies. In the `Update` method, add the following code after calculating the elapsed time:

C#

```
if (!IsAlive)
```



```
return;
```

Let's modify the `Draw` method to play the proper animation, depending on the state of the enemy. Replace the following code:

C#

```
if (!Level.Player.IsAlive ||
    Level.ReachedExit ||
    Level.TimeRemaining == TimeSpan.Zero ||
    waitTime > 0)
{
    sprite.PlayAnimation(idleAnimation);
}
else
{
    sprite.PlayAnimation(runAnimation);
}
```

with this code:

C#

```
if (!IsAlive)
{
    sprite.PlayAnimation(dieAnimation);
}
else if (!Level.Player.IsAlive ||
    Level.ReachedExit ||
    Level.TimeRemaining == TimeSpan.Zero ||
    waitTime > 0)
{
    sprite.PlayAnimation(idleAnimation);
}
else
{
    sprite.PlayAnimation(runAnimation);
}
```

Finally, add the `OnKilled` method:

C#

```
public void OnKilled(Player killedBy)
{
    IsAlive = false;
    killedSound.Play();
}
```

That completes the modification of `Enemy.cs`.

All right, we are coming up on the home stretch now. There is one final file to modify before you can compile and check out the power-up gem in action.

Modifying the Level Class

The modifications for the `Level` class are pretty extensive. You'll need to add support for a new gem type, and also support for killing an enemy if the invincible character collides with it.

First, let's add support for the new gem type. In the `Tile LoadTile(char tileType, int x, int y)` method, modify the code for the 'G' case to match the following:

C#

```
// Gem
case 'G':
    return LoadGemTile(x, y, false);

// Power-up gem
case 'P':
    return LoadGemTile(x, y, true);
```

This updates the code for a normal gem, passing **false**, and adds the ability to read in power-up gems. Now, you'll need to update the `LoadGemTile` method to accept a new parameter. Modify `LoadGemTile` to accept an additional parameter (of type **bool**) called `isPowerUp`. Make sure that the new parameter is last in the method declaration. It should match the following declaration:

C#

```
private Tile LoadGemTile(int x, int y, bool isPowerUp)
```

In this same function, modify the call to `gem.Add` to match the following:

C#

```
gems.Add(new Gem(this, new Vector2(position.X, position.Y), isPowerUp));
```

This modification now calls the recently modified `Gem` constructor.

If you remember, you earlier changed the implementation of the point value for a gem. Now is a good time to update the `OnGemCollected` method to use the new `PointValue` field. In this method, modify the `score` assignment to match the following:

C#

```
score += gem.PointValue;
```

Now that you can recognize and load the power-up gem, let's add support for its effect when the player character collects it.

Modify the `UpdateEnemies` method to first check that the current enemy is alive and, if so, upon collision check for a powered up player. If the player is powered up, you need to "kill" the enemy.

Change this code:

C#

```
if (enemy.BoundingBox.Intersects(Player.BoundingBox))
{
    OnPlayerKilled(enemy);
}
```

to match the following:

C#

```
if (enemy.IsAlive && enemy.BoundingBox.Intersects(Player.BoundingBox))
{
    if (Player.IsPoweredUp)
    {
        OnEnemyKilled(enemy, Player);
    }
}
```

```
    }  
    else  
    {  
        OnPlayerKilled(enemy);  
    }  
}
```

Next, add the implementation of the `OnEnemyKilled` method, after the `UpdateEnemies` method:

C#

```
private void OnEnemyKilled(Enemy enemy, Player killedBy)  
{  
    enemy.OnKilled(killedBy);  
}
```

This follows the practice of Platformer. It implements an `OnX` method to announce an important event has occurred.

Testing the Power-Up Extension

Before you can see the power-up gem in action, you'll need to add one to the second level structure file. Open this file now (1.txt) and replace any G with a P. Recompile and check out the Invincibility power-up!



Figure 2. Invincible Player Character and a Dead Enemy

See Also

Concepts

[Starter Kit: Platformer](#)

[Intermediate Platformer Features](#)

Tasks

[Platformer: Adding a Scrolling Level](#)

Advanced Platformer Features

Discusses advanced features of the Platformer Starter Kit, and offers recommendations for modifying or extending Platformer features.

In this section, we'll explore Platformer's advanced features and the classes that implement them, such as the game levels and the player character. We'll talk about their purpose and design, and point out features that can be modified or expanded on. For information on basic classes and features, see [Exploring the Basic Classes](#) and for advanced classes and features, see [Advanced Platformer Features](#).

- [Exploring the Advanced Classes](#)
- [Loading the Level](#)
- [Exploring Platformer's Physics](#)
- [Expanding Platformer](#)



Figure 1. Gameplay Screen for the Platformer Starter Kit

Exploring the Advanced Classes

The following classes are more complex than the basic and intermediate classes, and they implement the critical features of Platformer such as the game level, the player character, and the Platformer game structure.

PlatformerGame Class

The `PlatformerGame` class implementation is located in `PlatformerGame.cs`. Important methods include:

LoadContent

Loads the game fonts and the next level of Platformer by calling `LoadNextLevel`, which is another very important method.

For more information on level loading, see [Exploring Platformer's Physics](#).

Update

Handles basic game input (for both keyboard and gamepad) and calls `Level.Update`, ... you guessed it, another very important method. For more information on level updating, see the `Level` class description.

Draw

Draws the major components of Platformer: the gameplay screen and the HUD. The gameplay screen is drawn by calling `Level.Draw`, which in turn draws some things and calls the `Draw` method of any level object smart enough to draw itself (for example, the player character).

The structure of the game is easy to follow because the majority of Platformer objects are responsible for loading, updating, and drawing themselves. The `PlatformerGame` class is like a conductor; it tells each object what to do and when. As discussed in the [overview](#), the program flow can be broken up into three common steps: initializing (loading of the level, etc.), updating, and drawing. Each of these stages is discussed in detail in the appropriate class sections.

The `LoadNextLevel` method is an important step in the process of loading a level. This method locates and reads a physical file describing the current level. Level files are numbered sequentially, starting with 0, and stored in the `..\Content\Levels` directory (in the appropriate platform/build directory). For instance, the Level 0 file for the debug Windows version of Platformer is located in the `..\Platformer\Platformer\bin\x86\Debug\Content\Levels\0` directory.

⚠ Caution

Platformer ships with some ready-to-use levels. If you add new levels, ensure that the **Copy to Output Directory** property, for the related game asset, is set to **true**. In addition, follow the naming convention and directory structure for new levels. For instance, a new level would be in the 2 directory, underneath Levels, in the appropriate game content directory. For more information on the level structure file, see [Loading the Tiles](#).

`PlatformerGame` tracks the current level using the `levelIndex` variable. When `LoadNextLevel` is called, `levelIndex` is pre-incremented and an absolute path is built to the file. Once the directory is found, the file is passed to the `Level` constructor and the level is created and populated. If the level needs to be reloaded (due to running out of time), `ReloadCurrentLevel` is called.

Level Class

The `Level` class implementation is located in `Level.cs`. Important methods include:

Draw

Draws the current state of the level and its contained elements. Level tiles are drawn by calling `DrawTiles`. The player character, gems, and enemies are drawn by calling their respective `Draw` methods.

Each level has three layers of background. These backgrounds are loaded in the `Level` constructor. They are drawn in order, from back to front. After all layers have been drawn, the level tiles, player character, gems, and enemies are drawn. The base implementation of Platformer uses only the leftmost segment of the background texture assets.

Tip

For a tutorial on adding parallax scrolling to Platformer, see [Platformer: Adding a Scrolling Background](#).

Update

Updates the level state, including all objects contained by the level. In addition, collision checking between objects (such as the player character and platform tiles) and the time limit for the level (includes remaining time animation) are also handled here.

While the game is paused (either through a player death or reaching the exit), all input is ignored and the Idle animation for the enemies is played.

LoadTiles

Called during level construction, this method loads the level layout using a structure file previously located by `PlatformerGame.LoadNextLevel`. For more information on this level structure file, see [Loading the Tiles](#).

OnExitReached, OnPlayerKilled, and OnGemCollected

Called from `Update` when important events occur in a level. These events include the player character reaching the exit, dying, or collecting a gem. Modify these to change the default behavior of these events.

Player Class

The `Player` class implementation is located in `Player.cs`. Important methods include:

ApplyPhysics

Called during `Player.Update`, it updates the player's velocity and position based on input, gravity, and drag factors. Once position is determined, the player character is checked for collision with surrounding tiles. This is the most math-intensive method of the class, and it depends heavily on the game world physics model.

Draw

Orients player correctly, and then calls `AnimationPlayer.Draw` to draw the current frame of the character. As mentioned elsewhere, animation frames are intentionally square. This simplifies the animation code, making it easier to understand.

Update

Gets input from either the keyboard or a game controller, and updates the `movement` and `isJumping` fields. The player's position and velocity are then updated by calling `ApplyPhysics`.

LoadContent

Loads the various sprite sheets and sounds used by the player character.

HandleCollisions

Implements collision detection between the player character and neighboring tiles.

Important properties include:

BoundingBox

Used for checking collisions between the player character and neighboring tiles and objects, like a gem or exit location. The `RectangleExtensions` class is an important part of the collision checking code. For more information, see [RectangleExtensions](#) class.

Tip

Consider using per-pixel collision detection to improve the interaction between the player character and the level elements. This is an advanced extension, but it is worth the effort. For more information, download the 2D Per-Pixel Collision sample found on the [XNA Creators Club Web site](#). It's part of Collision Series 2.

Velocity

The velocity of the player character. This is affected by the game world physics, and it represents acceleration from running, jumping, or falling. For more information, see [Platformer Physics](#).

RectangleExtensions Class

This class is an extension to the XNA Framework [Rectangle](#) structure. `RectangleExtensions` implements two methods – `GetIntersectionDepth` and `GetBottomCenter`. These methods support the conceptualization of a rectangle as a "squared circle" with two radii.

Here is what I mean. It is given that a circle has one radius; the length from the center to its perimeter. This makes intersection checking with another circle easy. If the distance between the centers is shorter than the sum of the radii, the circles intersect.

Now apply this concept to a rectangle. Imagine that a rectangle has two radii: one from the center to the height of the rectangle (the "height" radius) and another from the center to the width of the rectangle (the "width" radius). Once you know the centers of the rectangles, you can use a similar approach to check for rectangle intersection as you did earlier with the two circles. If the X-value of the distance between the centers is greater than the sum of the width radii and the Y-value of the distance between the centers is greater than the sum of the height radii, the rectangles do not intersect.

Loading the Level

When starting a new game, the procedure for loading a level (from a program execution viewpoint) is as follows:

1. `PlatformerGame.LoadContent`
2. `PlatformerGame.LoadNextLevel`
3. `Level.Level`
4. `Level.LoadTiles`

A level can also be loaded in the following cases:

- When the player character reaches the exit of the current level. The next level is then loaded from `PlatformerGame.HandleInput` when the player continues the game.
- When reloading the level. This is called if time runs out. These events cause a restart of the level.

Before `PlatformerGame` can load the level, it must first find a physical file that contains the level structure. The `LoadNextLevel` method looks in the `..\Content\Levels` directory (underneath the appropriate platform/build directory) for these level structure files. Once found, a new level is constructed with a call to `Level.Level`.

In the level constructor, the level time is initialized, the tiles are loaded (using the level structure file) and then the background textures and "exit reached" sound are loaded. The `LoadTiles` method is the important part here.

Loading the Tiles

This is the workhorse of the level loading process. This method first reads in the character-based level file and verifies that all lines are the proper length. It then iterates over each line, reading the level structure from top to bottom until complete. A 2D array (composed of `Tile` objects) is then allocated to store each tile in the level.

At this point, the text-based lines, read from the level structure file, can be converted into actual tile layers for the current level. The `LoadTile` method reads a character from the current line and, based on a dictionary of characters, loads the appropriate type and game assets for the tile. The dictionary for the level structure file is as follows:

Text Character	Description
'.'	Blank tile. This tile is <code>TileCollision.Passable</code> , and has no related texture.
'X'	Exit tile. This tile is <code>TileCollision.Passable</code> , and uses the <code>Exit.png</code> texture. The location is stored, and then the tile assets are loaded with a call to <code>Level.LoadTile</code> .
'G'	Gem tile. This tile is <code>TileCollision.Passable</code> , uses the <code>Gem.png</code> texture, and loads the tile by calling <code>LoadGemTile</code> . The base implementation of <code>Platformer</code> adds a gem to the list of gems maintained by the level, and places the gem in an blank tile.

'-'	Floating platform. This tile is <code>TileCollision.Platform</code> , and uses the <code>Platform.png</code> texture.
'A', 'B', 'C', and 'D'	Starting tile for the various Platformer enemies. This tile is <code>TileCollision.Passable</code> , and contains one of the four enemies used by Platformer. They differ only in appearance; an obvious place for extending the base implementation of Platformer! The base implementation of Platformer instantiates an <code>Enemy</code> object (adding the enemy to the level's <code>enemies</code> list), and places it in the level.
'~'	Platform block. This tile is <code>TileCollision.Platform</code> , and uses a randomly-chosen texture from the set of textures whose name begins with "BlockB." This tile is loaded by calling <code>LoadVarietyTile</code> , which is a method that randomly loads a tile texture from a subset of tile textures.
':'	Passable block. This tile is <code>TileCollision.Passable</code> , and uses a randomly-chosen texture from the set of textures whose name begins with "BlockB."
'1'	Starting position for Player 1. This tile is <code>TileCollision.Passable</code> , and contains the player character. The base implementation of Platformer instantiates a <code>Player</code> object, and places him, facing the player, in the level. The location in the level is stored and used, in the event of player death, as a resurrection point.
'#'	Impassable block. This tile is <code>TileCollision.Impassable</code> , and uses a randomly-chosen texture from the set of textures whose name begins with "BlockA."

Exploring Platformer's Physics

Platformer implements a basic game physics model. It is used to perform the various actions of the player character: jumping, running, and falling. The main implementation is provided by `ApplyPhysics` and `GetJumpVelocity`. Both are implemented in the `Player.cs` source file.

ApplyPhysics

This method contains the bulk of the physics modeling, and is only called from `Level.Update`. It is responsible for computing the base velocity of the player character using horizontal movement and downward acceleration (gravity). In addition, pseudo drag is applied, gradually slowing the character down if he is running along a flat surface without input from the player. The player character also has a top running speed. Once the base velocity is computed, `GetJumpVelocity` is called.

GetJumpVelocity

This method is called from `ApplyPhysics` and computes the Y velocity (either jumping up or falling down) of the player character.

During jump ascension, the Y velocity is overridden by a power curve, as seen below.

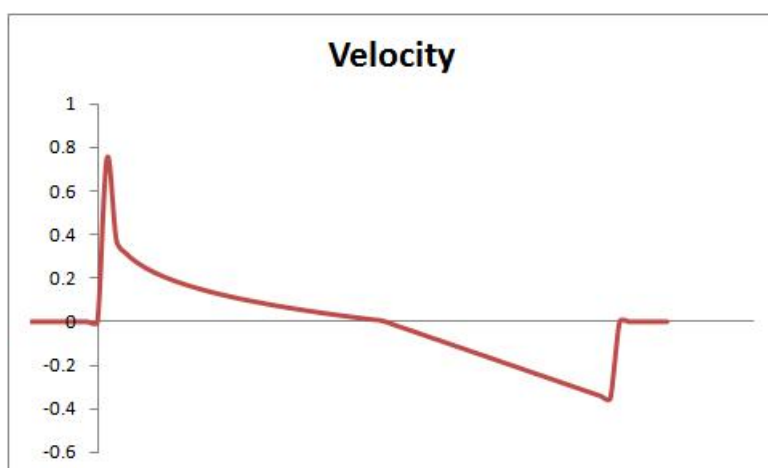


Figure 2. Velocity Curve (over Time)

During descent, gravity controls the Y velocity and clamps at a terminal velocity. The acceleration curve is graphed below.

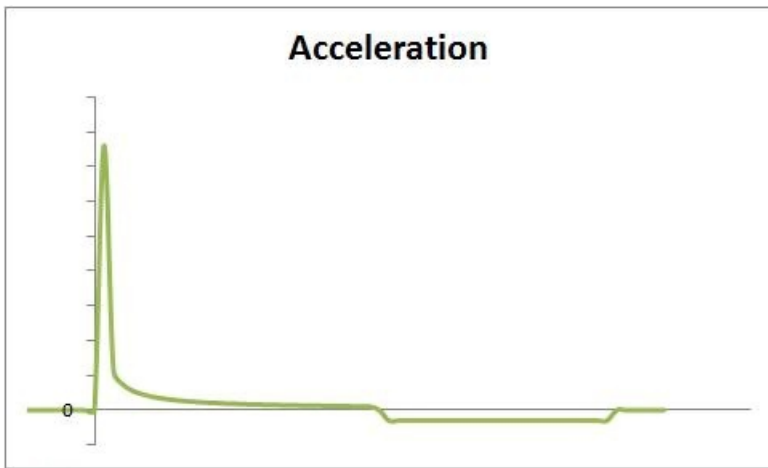


Figure 3. Acceleration Curve (over Time)

Expanding Platformer

A common feature of Platformer-type games is a scrolling level. In [Platformer: Adding a Scrolling Level](#), you'll add a scrolling level with multiple scrolling backgrounds to the default version of the Platformer starter kit.

See Also

Concepts

[Starter Kit: Platformer](#)

Tasks

[Platformer: Adding a Power-Up](#)

Platformer: Adding a Scrolling Level

Extends the base Platformer starter kit code by adding a scrolling level. Specifically, it provides parallax scrolling.

One of the more impressive features of platformer games in the 80s was the scrolling level. This topic details the process for adding this feature to the Platformer game. You'll implement multiple scrolling backgrounds, with the back layer scrolling the slowest, and the front layer scrolling the fastest. This is called parallax scrolling. It provides an illusion of depth in the game.



Figure 1. Parallax Scrolling Extension

Tip

It is *highly* recommended that you are already familiar with the structure and features of the Platformer starter kit. This extension involves modifications to several files in the Platformer starter kit. For more information on the Platformer starter kit, see [Starter Kit: Platformer](#).

This extension modifies two areas of the Platformer starter kit, and it adds a new class. It is recommended that you use the base Platformer starter kit solution as the starting point for your modifications.

Adding parallax scrolling involves the following major steps:

- Modifying `PlatformerGame.cs` to call `SpriteBatch.Begin` and `SpriteBatch.End` at a different time.
- Modifying `Level.cs` to use a new object type for the background textures instead of `Texture2D`. In addition, a camera is implemented and used to draw a portion of the background instead of the entire background.
- Adding a new class called `Layer` that replaces the usage of `Texture2D` for background textures.

Modifying the PlatformerGame Class

The only modification for this class involves modifying the `Draw` method. You'll move the `SpriteBatch.Begin` and `SpriteBatch.End` calls to the `DrawHud` method. This allows the `Level.Draw` method to set up its own batch for drawing the scrolling backgrounds.

In `PlatformerGame.Draw`, remove the `spriteBatch.Begin()`; and `spriteBatch.Begin()`; lines of code.

In the `PlatformerGame.DrawHud` method, add the following line before any existing code:

C#

```
spriteBatch.Begin();
```

At the end of the same method, add:

C#

```
spriteBatch.End();
```

The `DrawHud` method now implements a single batch.

That completes the modifications for `PlatformerGame.cs`. The next step adds support for a new class, called `Layer` to the `Level` class.

Modifying the Level Class

The main point of these modifications is to support the usage of a new kind of texture class (`Layer`) that enables parallax scrolling. Each of the three background textures will use the new class; therefore, the surrounding code also needs to accommodate the new class.

First, change the type used by the `layers` array from `Texture2D[]` to `Layer[]`. This is the new background texture class, added later.

C#

```
private Layer[] layers;
```

Now, look for a variable block, commented as "Level game state," and add a new variable called `cameraPosition`.

C#

```
private float cameraPosition;
```

This represents the leftmost world coordinate that can be seen by the player. Typically, a 3D camera sits at some position in world space and has a field of view. In the case of `Platformer`, the camera is represented by the left-most coordinate in the field of view and the field of view is inferred from the width of the viewport.

In the `Level` constructor, you'll need to change the initialization code for the `layers` array. Replace this code:

C#

```
layers = new Texture2D[3];
for (int i = 0; i < layers.Length; ++i)
{
    // Choose a random segment if each background layer for level variety.
    int segmentIndex = random.Next(3);
    layers[i] = Content.Load<Texture2D>("Backgrounds/Layer" + i + "_" + segmentIndex);
}
```

with the following:

C#

```
layers = new Layer[3];
layers[0] = new Layer(Content, "Backgrounds/Layer0", 0.2f);
layers[1] = new Layer(Content, "Backgrounds/Layer1", 0.5f);
layers[2] = new Layer(Content, "Backgrounds/Layer2", 0.8f);
```

The new code initializes the array with three new `Layer` objects. Each of these objects loads a different texture, and has a different scrolling speed (the third parameter of the `Layer` constructor).

Note

`Platformer` assumes that scrolling speed values have a range between 0 and 1. A value of 0 means no scrolling and 1 means scrolling at the same pace as the level tiles.

It's now time to modify the drawing code for the level. Locate the `Level.Draw` method, and replace it with the following method declaration:

C#

```
public void Draw(GameTime gameTime, SpriteBatch spriteBatch)
{
    spriteBatch.Begin();
    for (int i = 0; i <= EntityLayer; ++i)
        layers[i].Draw(spriteBatch, cameraPosition);
    spriteBatch.End();

    ScrollCamera(spriteBatch.GraphicsDevice.Viewport);
    Matrix cameraTransform = Matrix.CreateTranslation(-cameraPosition, 0.0f, 0.0f);
    spriteBatch.Begin(SpriteBlendMode.AlphaBlend, SpriteSortMode.Immediate, SaveStateMode
        .None, cameraTransform);

    DrawTiles(spriteBatch);

    foreach (Gem gem in gems)
        gem.Draw(gameTime, spriteBatch);

    Player.Draw(gameTime, spriteBatch);

    foreach (Enemy enemy in enemies)
        enemy.Draw(gameTime, spriteBatch);

    spriteBatch.End();

    spriteBatch.Begin();
    for (int i = EntityLayer + 1; i < layers.Length; ++i)
        layers[i].Draw(spriteBatch, cameraPosition);
    spriteBatch.End();
}
```

Now, let's go over what just changed. Initially, the first sprite batch draws all three background layers. Then, instead of moving a camera throughout the world, you'll move the world backwards such that the camera is always at the origin. This greatly simplifies the drawing logic because you can now call a specialized `SpriteBatch.Begin` overload that uses a transform matrix (calculated earlier in the method).

You'll recognize the next chunk of code because it is unchanged from the original implementation. It draws the level elements: tiles, gems, enemies, and the player character. The last batch does nothing in the base implementation of `Platformer`. It is left in for drawing foreground textures. For example, if a foreground texture (such as trees or bushes) was drawn, it would obscure the player character when he "walked" behind the texture.

Because the scrolling extension draws tiles off-screen, you should be aware that this could impact the frame rate. To avoid any slowdown you'll need to implement a simple culling feature that limits the amount of tiles drawn to only those on the screen at the time. This reduces the drawing load, speeding up the game. At the beginning of the `DrawTiles` method, add the following code:

C#

```
// Calculate the visible range of tiles.
int left = (int)Math.Floor(cameraPosition / Tile.Width);
int right = left + spriteBatch.GraphicsDevice.Viewport.Width / Tile.Width;
right = Math.Min(right, Width - 1);
```

Below this modification, modify the first line of the inner loop (the one that loops from 0 to `Width`) to match the following:

C#

```
for (int x = left; x <= right; ++x)
```

Now, only visible tiles are drawn but note that other items, such as gems and enemies, are still drawn even when off screen. The culling of non-tiles is another excellent place for extending Platformer!

The last modification in this file adds the new `ScrollCamera` method. This method calculates how much background is scrolled when the player reaches the screen's edge. When the begin scrolling is platform-dependent. Because the Zune screen is the narrowest of the three, it looks the farthest ahead. The other two don't look ahead as much. This factor is used to calculate the edges of the screen and how far to scroll when the player reaches that edge. Scrolling continues until either end of the level is reached. At that point, the camera position is clamped.

Add the following code, after the `Draw` method:

```
C#  
  
    private void ScrollCamera(Viewport viewport)  
    {  
#if ZUNE  
const float ViewMargin = 0.45f;  
#else  
const float ViewMargin = 0.35f;  
#endif  
  
    // Calculate the edges of the screen.  
float marginWidth = viewport.Width * ViewMargin;  
float marginLeft = cameraPosition + marginWidth;  
float marginRight = cameraPosition + viewport.Width - marginWidth;  
  
    // Calculate how far to scroll when the player is near the edges of the screen.  
float cameraMovement = 0.0f;  
if (Player.Position.X < marginLeft)  
    cameraMovement = Player.Position.X - marginLeft;  
else if (Player.Position.X > marginRight)  
    cameraMovement = Player.Position.X - marginRight;  
  
    // Update the camera position, but prevent scrolling off the ends of the level.  
float maxCameraPosition = Tile.Width * Width - viewport.Width;  
cameraPosition = MathHelper.Clamp(cameraPosition + cameraMovement, 0.0f, maxCameraPos  
ition);  
    }  
}
```

The final step adds the new `Layer` class.

Implementing the Layer Class

Because the backgrounds will be scrolling during gameplay, you'll need something more specialized than a `Texture2D` class to draw these textures. The background textures provided are divided into three segments that tile seamlessly into one scrolling background.

Using the **Add Class** dialog, add a new C# class, called `Layer`, to the PlatformerWindows solution. At the top of the file, add some useful XNA Framework references:

```
C#  
  
using Microsoft.Xna.Framework;  
using Microsoft.Xna.Framework.Graphics;  
using Microsoft.Xna.Framework.Content;
```

In this new class, add the following properties:

```
C#  
  
public Texture2D[] Textures { get; private set; }
```

```
public float ScrollRate { get; private set; }
```

These properties store the background texture of the layer and its scroll speed.

Now add the constructor:

C#

```
public Layer(ContentManager content, string basePath, float scrollRate)
{
    // Assumes each layer only has 3 segments.
    Textures = new Texture2D[3];
    for (int i = 0; i < 3; ++i)
        Textures[i] = content.Load<Texture2D>(basePath + "_" + i);

    ScrollRate = scrollRate;
}
```

This constructor accepts a content manager, a base path to the background asset, and the scroll speed of the background layer. Note that each layer has only three segments.

It loads each segment of the background in the `Textures` array, and then sets the scroll speed.

The final method to add is the `Draw` method. Add this code after the constructor method:

C#

```
public void Draw(SpriteBatch spriteBatch, float cameraPosition)
{
    // Assume each segment is the same width.
    int segmentWidth = Textures[0].Width;

    // Calculate which segments to draw and how much to offset them.
    float x = cameraPosition * ScrollRate;
    int leftSegment = (int)Math.Floor(x / segmentWidth);
    int rightSegment = leftSegment + 1;
    x = (x / segmentWidth - leftSegment) * -segmentWidth;

    spriteBatch.Draw(Textures[leftSegment % Textures.Length], new Vector2(x, 0.0f), Color.White);
    spriteBatch.Draw(Textures[rightSegment % Textures.Length], new Vector2(x + segmentWidth, 0.0f), Color.White);
}
```

This method first calculates which of the background segments to draw, and then draws them offset by the previously calculated amount. It is assumed that two segments are enough to cover any screen.

Modifying the Level Structure File

At this point, the parallax scrolling extension is completely coded. However, if you recompile and run the game, you will see no difference. Like the Power-Up Gem [extension](#), you'll need to modify an existing map to enable the scrolling. However, it won't be as easy as modifying an extra character or two. You would need to come up with a lot of new content, past the default edge of the level, to fully illustrate the scrolling. However, since the format of the level structure file is text-based, you can just use the following text block as a demonstration case.

C#

```
.....
.....
.....G.....X...
.....###.....#####
.....G.....
```

```

.....###.....G.GDG.....G.G.G.....
.....G.....#####.....#####.....
.....###.....
.....G.....G.G.....G.G.....
.....###.....#####.....#####.....
.....G.....
.....###.....GDG.G.....G.G.G.G.G.G.G.G.G.
.....#####.....##.G.G.G.G.G.G.G.G..
.1.....GCG.G.G.GCG.G.G.GCG.
#####.....#####

```

Copy this text and then open the 0.txt file (located in the HighResolutionContent content project). Select all text in the level structure file, and then paste the new text in. Save the file and do the same for the equivalent low-resolution level structure file. Once you have made changes to both maps, recompile and run the game. You can now run far to the right in the first level, and the three background layers scroll at different speeds. Pretty cool effect, eh?!

See Also

Concepts

[Starter Kit: Platformer](#)

[Advanced Platformer Features](#)

Tasks

[Platformer: Adding a Power-Up](#)

Platformer: Adding Touch Support

Extends the base Platformer starter kit code by adding touch screen and accelerometer support for input.

This extension modifies five areas of the Platformer starter kit. The following steps assume that you have generated a new Platformer starter kit project and it is loaded into XNA Game Studio.

Tip

To prevent build warnings and errors, it is recommended that you remove both the Windows and Xbox 360 versions of the project from the newly-generated Platformer solution. It is also recommended that you set the solution platform to Zune.

With these changes in place, you can now build and debug the starter kit without misleading errors and warnings.

Adding touch screen and accelerometer support involves the following major steps:

- Modifying PlatformerGame.cs to center the gameplay screen and accept touch screen input.
- Modifying Level.cs to support input from the accelerometer and touch panel.
- Modifying Player.cs to control the player character using input from the accelerometer and touch screen.
- Modifying Enemy.cs to offset the enemy location. This accounts for the overall offsetting of the gameplay screen required by the higher resolution of a Zune HD device.
- Modifying Gem.cs to offset the gem location. This accounts for the overall offsetting of the gameplay screen required by the higher resolution of a Zune HD device.

For more information on touch screen and accelerometer support, see [Zune HD Input Overview](#).

Modifying PlatformerGame.cs

The original Platformer starter kit was designed for Zune devices with a resolution of 240 x 320. However, the Zune HD device supports a higher resolution: 272 x 480. If you were to install an unmodified version of Platformer on the new Zune, two things would be obvious immediately:

- The gameplay screen does not fill the entire screen. The right edge of the level is not flush with the screen edge, and a large portion of the default cornflower blue screen is visible on the bottom.
- There is no way to control the player character. Unlike the earlier Zune device, the Zune pad is gone and there is no dedicated Back button.

The first set of modifications you'll make addresses the higher resolution issue. First, the preferred back buffer dimensions will be increased, and then an offset will be used to shift the gameplay screen over and down. This offset centers gameplay much like letterboxing centers a 4:3 image on a widescreen television. After the modification, you will see thin black bars (on the sides) and thick black bars (on the top and bottom).

Open the PlatformerGame.cs file and, near the top of the file, modify the width and height of the back buffer to match the following:

C#

```
private const int BackBufferWidth = 272;
private const int BackBufferHeight = 480;
```

The next modification changes the default cornflower blue screen color to black. This supports the illusion of letter boxing. In the Draw method, modify the [Clear](#) method call to match the following:

C#

```
graphics.GraphicsDevice.Clear(Color.Black);
```

The final modification uses a 2D vector to offset the text drawn on the gameplay screen. In the same file, locate the `DrawHud` method and declare a new [Vector2](#) object with a horizontal and vertical offset at the top of the method:

C#

```
Vector2 screenOffset = new Vector2(16, 80);
```

This vector is added to the current position of any shadowed text, causing the text to be drawn farther to the right (half the difference between the old and new screen width) and farther down (half the difference between the old and new screen height). Let's modify those placements now. In the same method, find the two calls to `DrawShadowedString`. Modify the third parameter (`hudLocation`) by adding the new offset vector to the current value. The code below is the result of modifying the second call:

C#

```
DrawShadowedString(hudFont, "SCORE: " + level.Score.ToString(),
    hudLocation + screenOffset
    + new Vector2(0.0f, timeHeight * 1.2f), Color.Yellow);
```

That completes the modifications needed to center the gameplay screen on the Zune device. The next modifications focus on adding touch screen support for responding to game status messages.

First, let's add support for the state of the device's touch screen and accelerometer. After adding the back buffer constants, insert the following code:

C#

```
AccelerometerState accelState;
TouchCollection touchState;
```

These states are used mainly by the `PlatformerGame` and `Player` objects. Due to the nature of the touch screen and accelerometer states, you should only retrieve these device states once per frame. Then you pass the current state value (either `accelState` or `touchState`) to other objects that need input from those devices. It is not recommended that you call either of the `GetState` methods more than once per frame as different values will result.

In the same file, locate the `Update` method, and modify the call to `Level.Update` to pass the states of the accelerometer and touch screen to the `Level` object:

C#

```
level.Update(gameTime, accelState, touchState);
```

Now that the device states have been passed on, you'll add new code to the `HandleInput` method that responds to user touches on the screen when a status overlay message is displayed (game win or loss). The first step is to retrieve the states of the touch screen and accelerometer with calls to the `GetState` method. Add the following code after the existing `GetState` calls:

C#

```
accelState = Accelerometer.GetState();
touchState = TouchPanel.GetState();
bool buttonTouched = false;
```

The next modification looks at the collection of touch locations and checks each location for a `TouchLocation.Pressed` state. If one is found, the `buttonTouched` variable is updated to `true`. The code used for this check is fairly common when querying the state of the current touch locations. For demonstration purposes, the three major states are checked, but the code only reacts to screen presses. For more information on using states, see [Zune HD Input Overview](#). Add the following code after the check for the exit condition:

C#

```
//interpret touch screen presses
foreach (TouchLocation location in touchState)
{
    switch (location.State)
    {
        case TouchLocationState.Pressed:
            buttonTouched = true;
            break;
        case TouchLocationState.Moved:
            break;
        case TouchLocationState.Released:
            break;
    }
}
```


Now that the `buttonTouched` variable holds the correct value, modify the `continuePressed` assignment to also use this value:

C#

```
bool continuePressed =
    keyboardState.IsKeyDown(Keys.Space) ||
    gamepadState.IsButtonDown(ContinueButton) || buttonTouched;
```

At this point, some of the gameplay screen elements are offset properly, and the game responds to touches when a status overlay message is displayed. It's now time to move on to the modification of the `Level.cs` file.

Modifying Level.cs

Modifications in this file consist mainly of applying position offsets to some elements. In addition, you'll update the `Level.Update` definition and pass those state values onto the `Player` object. That's the object that will mainly use these values.

First, add the `Microsoft.Xna.Framework.Input` namespace to the list of namespaces at the top of the file. This provides quick access to the new input-related types:

C#

```
using Microsoft.Xna.Framework.Input;
```

Locate the `Level.Update` method definition and modify it to match the following:

C#

```
public void Update(GameTime gameTime, AccelerometerState accelState, TouchCollection touchState)
```

Further down in that method, find the call to `Player.Update` and modify it to match the following:

C#

```
Player.Update(gameTime, accelState, touchState);
```

That completes the state-related modifications. The remaining changes offset more screen elements to further improve the letterbox appearance of the gameplay screen.

Locate the `Draw` method and add a screen offset vector declaration to the beginning of the function:

C#

```
Vector2 screenOffset = new Vector2(0, 80);
```

You can now use this offset to move the backgrounds to the center of the screen. Modify both `SpriteBatch.Draw` calls by adding the `screenOffset` vector value to the second parameter:

C#

```
spriteBatch.Draw(layers[i], Vector2.Zero + screenOffset,
    Color.White);
```

The next set of modifications are to the `DrawTiles` method, located after the `Draw` method. There is a slight difference between the screen offset vector used earlier. The vector used in the `DrawTiles` method also shifts the tiles to the right, centering them in the screen. This is necessary because the maps are slightly thinner than the width of the display.

Tip

The background textures, used by the Zune version of Platformer, have always been wider than the actual screen of the Zune device. Therefore, only the first portion of each background texture was seen even though the entire texture was loaded by the application. To achieve a solid edge to the gameplay screen (both tiles and background centered on the screen) you could modify the background layers of the `LowResolutionContent` project by specifying a width of 320 for each background layer. This is the original width, which matches the layout of the level tiles. However, if you do this, you will also have to change the value of the `screenOffset` vector used in the `Level.Update` method as follows: `new Vector2(16, 80);`.

Add a screen offset vector declaration to the beginning of the function:

C#

```
Vector2 screenOffset = new Vector2(16, 80);
```

Modify the existing `SpriteBatch.Draw` call by adding the `screenOffset` vector value to the second parameter:

C#

```
spriteBatch.Draw(texture, position + screenOffset,  
                Color.White);
```

At this point, the project doesn't compile (due to the `Player.Update` change), but the next step fixes that.

Modifying Player.cs

This is the main modification for the project. In this step, you'll modify the input code of the player character so that it responds to accelerometer and touch screen inputs. There are many ways to modify the original control schema. This extension takes the simple approach and uses a combination of both accelerometer and touch screen inputs. The player character is controlled by tilting the device. Tilt it to the left and the player character runs to the left edge of the screen; tilt it to the right and he runs to the right edge. Due to the sensitivity of the accelerometer there is a built-in dead zone that makes it easier to prevent the player character from constantly running back and forth. This can be modified to suit your individual preference. You could even expose this to the player as a customization feature. Finally, tapping the screen causes him to jump. The jump is determined by his velocity and current direction.

Open the `Player.cs` file and locate the `Update` method. You'll need to modify the declaration to match the call made in the `Level.cs` file. It should match the following:

C#

```
public void Update(GameTime gameTime, AccelerometerState accelState,  
                  TouchCollection touchState)
```

In the same method, modify the first line of code to match the following:

C#

```
GetInput(accelState, touchState);
```

This passes the current states of the accelerometer and touch screen so that the player character position and jumping state are properly updated. The next modification updates the `GetInput` method signature to match this new call, and it adds new code that controls the player character through touch and the tilt of the device.

Modify the first line of the `GetInput` method to match the following:

C#

```
private void GetInput(AccelerometerState accelState,  
                     TouchCollection touchState)
```

In the method body, find the location of the following comment: `// Check if the player wants to jump..` Just before this comment, insert the following code:

C#

```
if (Math.Abs(accelState.Acceleration.X) > 0.10f)  
{  
    if (accelState.Acceleration.X > 0.0f)  
        movement = 1.0f;  
    else  
        movement = -1.0f;  
}  
  
//override digital if touch input is found  
// Process touch locations.  
bool touchJump = false;
```

```

foreach (TouchLocation location in touchState)
{
    switch (location.State)
    {
        case TouchLocationState.Pressed:
            touchJump = true;
            break;
        case TouchLocationState.Moved:
            break;
        case TouchLocationState.Released:
            break;
    }
}

```

This code does two things: checks for accelerometer changes and for presses on the touch screen.

Player movement is controlled by the side-to-side tilt of the device. Tilting the device to the left turns and runs the character in that direction. Tilting to the right turns and moves the character to the right. Due to the sensitivity of the accelerometer it is necessary to code in a "dead zone" for the accelerometer input. The tilt value, along the x-axis, must be greater than 0.1 for the character to begin running. This allows small movement of the device along the x-axis without moving the character. If the acceleration value exceeds the threshold, the `movement` variable is updated with the proper value (positive for left movement, negative for right).

After the movement is determined, the touch screen is checked for any presses that occurred during the current frame. If any are found, the `touchJump` variable is set to **true**.

The final modification to this method is to add the `touchJump` value to the calculation of the `isJumping` variable. Modify the existing assignment to match the following:

C#

```

// Check if the player wants to jump.
isJumping =
    gamePadState.IsButtonDown(JumpButton) ||
    keyboardState.IsKeyDown(Keys.Space) ||
    keyboardState.IsKeyDown(Keys.Up) ||
    keyboardState.IsKeyDown(Keys.W) ||
    touchJump;

```

There is one more area to modify before you're done with this file: rendering of the player character. You can do this by modifying the existing `Player.Draw` method.

As with earlier modifications to the placement of game play elements, the player character also is offset using vector addition. The modifications are similar to those done in the past. Locate the `Draw` method, and add the following code to the beginning of the method:

C#

```

Vector2 screenOffset = new Vector2(16, 80);

```

Modify the existing `sprite.Draw` call to match the following:

C#

```

sprite.Draw(gameTime, spriteBatch, Position + screenOffset, flip);

```

This completes the `Player.cs` modifications. The remaining modifications finish the letterbox effect of the gameplay screen by shifting the enemies and gems in the current level.

Modifying Enemy.cs

In this file, the only code you need to modify is the `Enemy.Draw` method. Open the `Enemy.cs` file, and locate the `Draw` method. At the beginning of the method, add the following code:

C#

```

Vector2 screenOffset = new Vector2(16, 80);

```

Modify the last line of code in the method (the `Draw` method call) to match the following:

C#

```
sprite.Draw(gameTime, spriteBatch, Position + screenOffset, flip);
```

After modification, the enemy sprites are properly shifted on the gameplay screen to match the other game elements.

Modifying Gem.cs

As with the `Enemy.cs` file, the only modification being made is to the `Gem.Draw` method. Open the `Gem.cs` file and locate the `Draw` method. At the beginning of the method, add the following code:

C#

```
Vector2 screenOffset = new Vector2(16, 80);
```

Modify the `Draw` method call to match the following:

C#

```
spriteBatch.Draw(texture, Position + screenOffset, null, Color,
    0.0f, origin, 1.0f, SpriteEffects.None, 0.0f);
```

After modification, the gem sprites are properly shifted on the gameplay screen to match the other game elements.

See Also

Concepts

[Starter Kit: Platformer](#)

[Zune HD Input Overview](#)

Tasks

[Platformer: Adding a Power-Up](#)

[Platformer: Adding a Scrolling Level](#)

Support Options and Additional Resources

Setting Up Your Xbox 360 Console to Run Games Created with XNA Game Studio

For help with obtaining XNA Game Launcher, an XNA Creators Club membership, or in connecting your computer to an Xbox 360 console, contact Xbox 360 customer support.

1-800-4My-Xbox (1-800-469-9269)

<http://www.xbox.com/en-US/support/contact>

Xbox 360 customer support cannot provide help with developing games in XNA Game Studio.

Developing Games with XNA Game Studio

For help with developing games with XNA Game Studio, see the following additional resources.

XNA Creators Club Online	Your primary information source for developing games with XNA Game Studio.
XNA Creators Club Online Forums	Ask questions and exchange ideas with other game developers. The game development forums include forums specifically for XNA Game Studio, the XNA Framework, game design, and other topics.
XNA Developer Center on MSDN	The MSDN site for XNA Game Studio.
XNA Game Studio Blog	The XNA Game Studio Blog is a great place to keep up to date with the latest information from the XNA Game Studio team, including <ul style="list-style-type: none"> • What the team is working on, including feature roadmaps. • Insights into the people on the XNA Game Studio team. • Why and how decisions were made about the product. • Responses to questions and comments from the community. • Pointers to cool articles and information on XNA Game Studio and Xbox LIVE Indie Games. • Tips and tricks on using XNA Game Studio.
MSDN Library	The MSDN Library is an essential resource for developers using Microsoft tools, products, and technologies. It contains technical programming information, sample code, documentation, technical articles, and reference guides.

Reporting Issues and Making Suggestions

To report an issue with XNA Game Studio or the XNA Framework, or suggest how we might improve these products, use MS-Connect at the following location.

[Connect: XNA Game Studio](#)

Using XNA Game Studio

Describes how to use XNA Game Studio. XNA Game Studio is seamlessly integrated with supported versions of Microsoft Visual Studio tools, exposing an array of new and updated features for developing 2D and 3D games.

These features include:

- A game component model.
- A framework library for Microsoft Windows, Xbox 360, and Zune game development.
- Integration with the XNA Framework Content Pipeline.

XNA Game Studio brings these features together to provide an easier and quicker path for developing games for Windows, Xbox 360, and Zune.

This section discusses what you will need in order to develop managed game titles using the features of the XNA Framework.

In This Section

[Creating a Windows Game or Library Project](#)

Describes how XNA Game Studio includes a set of project templates that will help you to develop Windows-based projects using the XNA Framework.

[Developing Xbox 360 Games](#)

Describes how to develop games for the Xbox 360. Xbox 360 game development is similar to Windows game development with several notable differences.

[Developing Zune Games](#)

Describes how to develop games for Zune.

[Managing Game Assets](#)

Describes game assets, which are the collection of data files used to support gameplay such as bitmaps, models, textures, or sounds. Game assets are governed by the XNA Framework Content Pipeline.

[Developing Cross-Platform Games](#)

Describes how XNA Game Studio and the XNA Framework facilitate the development of games that work on several platforms.

[Game Studio Features](#)

Describes how XNA Game Studio provides numerous feature extensions to Visual Studio and Visual C# Express to help create eye-popping games.

[Sharing and Distributing Your Game](#)

Describes how to share and distribute your game to XNA Game Studio users and others.

See Also [XNA Game Studio 3.1](#)

Creating a Windows Game or Library Project

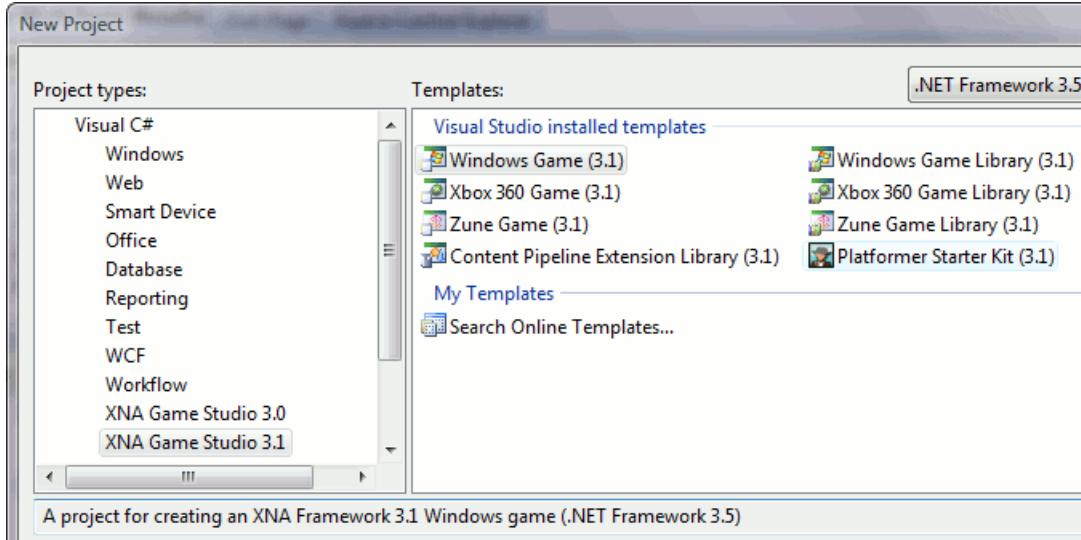
Describes how XNA Game Studio includes a set of project templates that will help you to develop Windows-based projects using the XNA Framework.

Starting a New Project

To start a new project

- To begin a new Windows-based project, click **File**, and then click **New Project**.

You'll be presented with a dialog box that lists a number of project templates.



XNA Game Studio provides template types for XNA Framework game development in the **XNA Game Studio 3.1** section of the **Visual C#** project types. The templates offered for developing XNA Framework games for Windows are:

- Windows Game (3.1)—A project for creating an XNA Framework 3.1 game application for Windows.
- Windows Game Library (3.1)—A project for creating an XNA Framework 3.1 game library for Windows.
- Content Pipeline Extension Library (3.1)—A project for creating an XNA Framework 3.1 Content Pipeline Extension Library.

For information about XNA Game Studio project templates for the Xbox 360 or Zune, see [Creating an Xbox 360 Game or Library Project](#) and [Creating a Zune Game or Library Project](#).

Note

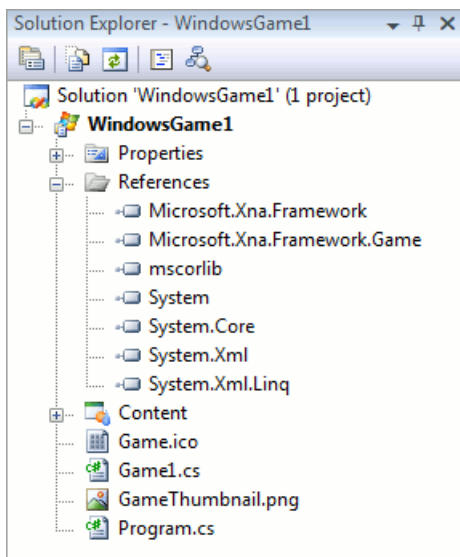
XNA Game Studio 3.1 continues to support creation and use of XNA Game Studio 3.0 projects. If you'd like to create a new XNA Game Studio 3.0 project, click the **XNA Game Studio 3.0** category in the **Project Types** box to gain access to the 3.0 project types. For more information, see [Upgrading XNA Game Studio Projects](#).

Windows Game (3.1)

XNA Game Studio provides a Windows Game (3.1) template that creates and loads a set of starter files. This new project contains basic code that implements a simple, framed window with a colored background.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can use the Project Designer to modify the values for these properties.

In addition to these properties, assembly information (such as the game title) is stored in the AssemblyInfo.cs file. You can use the [Assembly Information](#) dialog box to modify this information, or you can manually edit the .cs file.

References

References to the following assemblies are added automatically to a new Windows Game project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Game
- mscorlib
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

Content

The nested content project stores and builds content for the game. For more information, see [Game Content Project](#).

Game.ico File

A 32 × 32 default icon, representing your game.

Game1.cs File

This file is a good starting point for adding simple game logic and basic features. It implements a single class (derived from **Game** and called `Game1`), and it overrides five methods: **LoadContent**, **UnloadContent**, **Initialize**, **Draw**, and **Update**. In addition, the `Game1` constructor is defined. Use these methods to initialize your game components, load and render your game content, and handle any input from the user or changes to the game environment.

GameThumbnail.png

The icon appears when this game is packed for distribution as a .ccgame, which is described in [Sharing Your Game Package](#).

Program.cs File

This file also implements a single class (called `Program`) that provides an entry point to game execution. Usually, little code is added to this file unless the game is fairly advanced.

To create a Windows Game (3.1) project

1. On the **File** menu, click **New Project**.
2. Select the **Windows Game (3.1)** project type.
3. Type the name for the game project in the **Name** text box.

You can also modify the default values for the **Location** and **Solution Name** controls.

4. Click **OK** to create and load the new project.

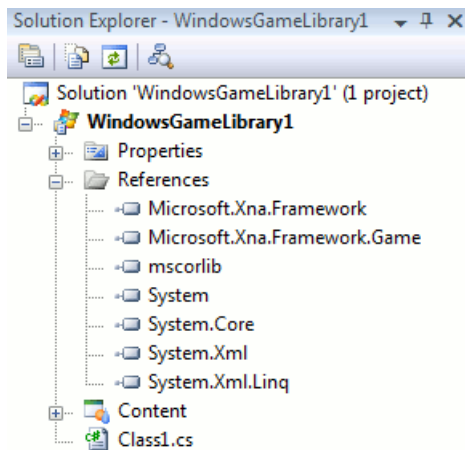
Windows Game Library (3.1)

XNA Game Studio provides a Windows Game Library (3.1) template that creates and loads a set of starter files. Typically, projects of this type contain managed classes that implement basic or advanced features used by a game engine or 3D renderer. Once completed, these class libraries can be referenced by other Windows projects. They provide common functionality without having that code reside within the game project.

The new project contains basic code that implements an empty library, usable by other Windows Game projects or Windows Game libraries.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can modify the values for these properties using the Project Designer.

In addition to these properties, assembly information (such as the library title) is stored in the AssemblyInfo.cs file. This information can be modified using the [Assembly Information](#) dialog box or by manually editing the .cs file.

References

References to the following assemblies are added automatically to a new Windows Game Library project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Game
- mscorlib
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

Content

The nested Content project stores and builds content that is to be included in the game library. This is useful for including reusable content assets that support the classes and methods in the game library. For more information, see [Game Content Project](#).

Class1.cs File

This file implements an empty C# class within a namespace. This provides a starting point for the class library.

To create a Windows Game Library (3.1) project

1. On the **File** menu, click **New Project**.
2. Select the **Windows Game Library (3.1)** project type.

3. Type the name for the library project in the **Name** text box.

You can also modify the default values for the **Location** and **Solution Name** controls.

4. Click **OK** to create and load the new project.

Content Pipeline Extension Library (3.1)

XNA Game Studio provides a Content Pipeline Extension Library (3.1) template that creates and loads a set of starter files. The template provides a starting point for developers wishing to create their own custom importers and processors for their game content. For more information on the XNA Framework Content Pipeline, and content processors in general, see [Content Pipeline](#).

Note

Content Pipeline Extension Library projects require .NET Framework 3.5. This version of the .NET Framework must be selected in the combo box in the upper-right corner of the **New Project** dialog box. Otherwise, the Content Pipeline Extension Library (3.1) template will not be presented. The .NET Framework 3.5 is the default selection.

The new project contains basic code that implements an empty processor, usable by the XNA Framework Content Pipeline. Solution configurations for all available platforms are created automatically. This means that the Content Pipeline extension project builds processors and importers for any available platform without any change to the project.

Note

A solution configuration determines which project configurations are active, as well as which projects are built and/or deployed when you invoke the **Build** and **Run** commands. By default, if a project does not have a configuration that matches the active solution configuration, the solution build will skip that project.

In addition to the project template, there are additional code templates related to projects of this type. You can use the **Add New Item** dialog box to add these items to your project. Here are some basic descriptions:

Content Processor item

An item template comprised of a single code file (called **ContentProcessorN.cs**). The numerical suffix (*N*) guarantees uniqueness within the project. This file is similar to the **ContentProcessor1.cs** file created for the Content Pipeline Extension Library (3.1) template. The file declares a class derived from [ContentProcessor](#) with an override of the abstract [Process](#) method.

Content Importer item

An item template comprised of a single code file called **ContentImporterN.cs**. The numerical suffix (*N*) guarantees uniqueness within the project. The file declares a class derived from [ContentImporter](#) with an override of the abstract [Import](#) method.

Content Type Writer item

An item template comprised of a single code file called **ContentTypeWriterN.cs**. The numerical suffix (*N*) guarantees uniqueness within the project. The file declares a class derived from [ContentTypeWriter](#) with overrides of the abstract [Write](#) and [GetRuntimeReader](#) methods.

Content Type Reader item

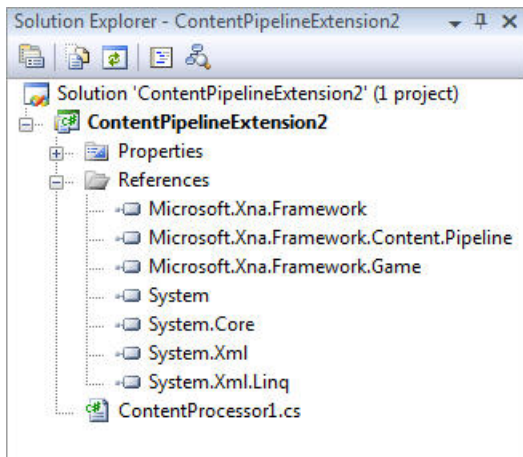
An item template comprised of a single code file called **ContentTypeReaderN.cs**. The numerical suffix (*N*) guarantees uniqueness within the project. The file declares a class derived from [ContentTypeReader](#) with an override of the abstract [Read](#) method.

Unlike the **Content Type Writer** item, this item is not available when adding a new item to a Content Pipeline Extensions Library project. It is only available for game or game library projects.

Whenever items of this type are added to a project, references to any dependent assemblies are added automatically. These references include assemblies that are likely to be required when customizing the newly-added class.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can modify the values for these properties using the Project Designer.

In addition to these properties, assembly information (such as the library title) is stored in the AssemblyInfo.cs file. This information can be modified using the [Assembly Information](#) dialog box or manually editing the .cs file.

References

References to the following assemblies are added automatically to a new Content Pipeline Extension Library project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Content.Pipeline
- Microsoft.Xna.Framework.Game
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

ContentProcessor1.cs File

This file implements an empty C# processor within a namespace, providing a starting point for the custom content processor.

To create a Content Pipeline Extension Library (3.1) project

1. On the **File** menu, click **New Project**.
2. Select the **Content Pipeline Extension Library (3.1)** project type.
3. Type the name for the library project in the **Name** text box.
You can also modify the default values for the **Location** and **Solution Name** controls.
4. Click **OK** to create and load the new project.

See Also [Using XNA Game Studio](#)
[Sharing Your Game Package](#)

Developing Xbox 360 Games

Describes how to develop games for the Xbox 360. Xbox 360 game development is similar to Windows game development with several notable differences.

This section discusses developing Xbox 360 game titles using the XNA Game Studio development environment.

In This Section

[Creating an Xbox 360 Game or Library Project](#)

Describes how XNA Game Studio includes a set of project templates that will help you to develop game projects for Xbox 360 using the XNA Framework.

[Deploying an Xbox 360 Game](#)

Describes how XNA Game Studio can copy executable and media files to the Xbox 360 retail console once you are ready to deploy them.

[Debugging an Xbox 360 Game](#)

Discusses issues that apply to debugging Xbox 360 games on your retail console.

[Troubleshooting Xbox 360 Game Deployment](#)

Provides some helpful troubleshooting tips for Xbox 360 deployment if you are unable to establish a successful connection between your computer and your Xbox 360 console.

See Also [Xbox 360 Programming Considerations](#)

[Using XNA Game Studio](#)

[Third-Party Firewall Settings](#)

Creating an Xbox 360 Game or Library Project

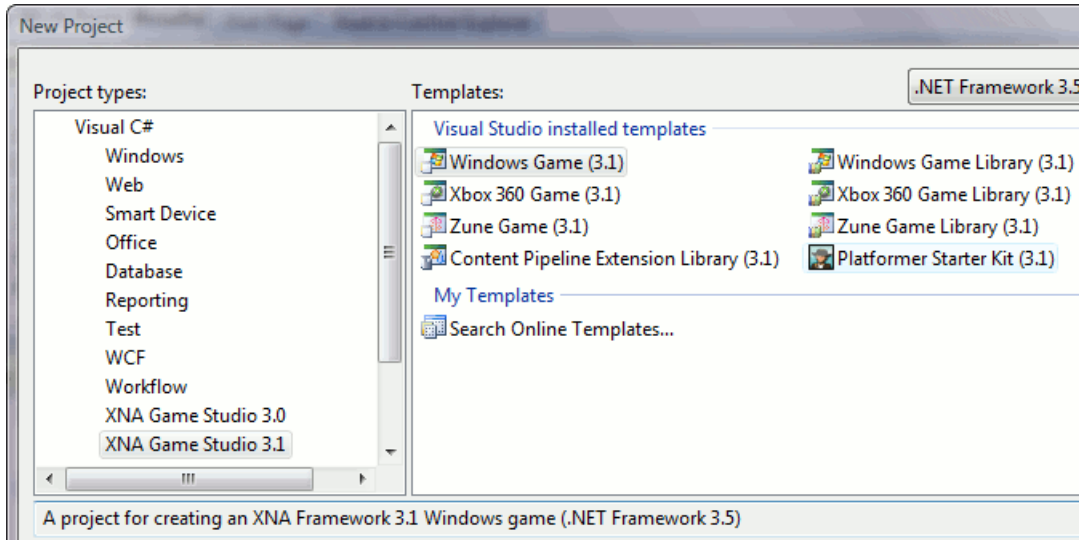
Describes how XNA Game Studio includes a set of project templates that will help you to develop game projects for Xbox 360 using the XNA Framework.

Starting a New Project

To start a new project

- To begin a new Xbox 360 project, click **File**, and then click **New Project**.

You'll be presented with a dialog box that lists a number of project templates.



XNA Game Studio provides template types for XNA Framework game development in the **XNA Game Studio 3.1** section of the **Visual C#** project types. The templates offered for developing XNA Framework games for Xbox 360 are:

- Xbox 360 Game (3.1)—A project for creating an XNA Framework 3.1 game application for Xbox 360.
- Xbox 360 Game Library (3.1)—A project for creating an XNA Framework 3.1 game library for Xbox 360.
- Content Pipeline Extension Library (3.1)—A project for creating an XNA Framework 3.1 Content Pipeline Extension Library. For more information, see [Creating a Windows Game or Library Project](#).

For information about XNA Game Studio project templates for Windows or Zune, see [Creating a Windows Game or Library Project](#) and [Creating a Zune Game or Library Project](#).

Note

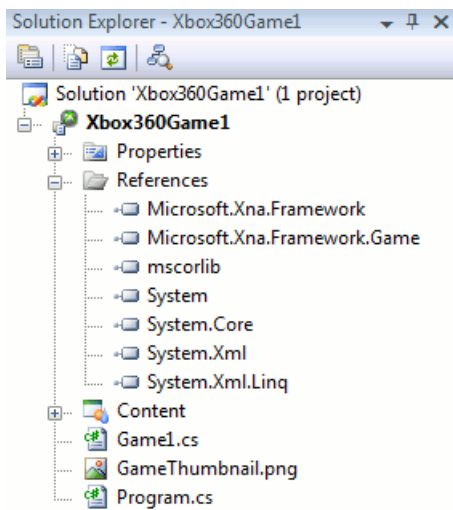
XNA Game Studio 3.1 continues to support creation and use of XNA Game Studio 3.0 projects. If you'd like to create a new XNA Game Studio 3.0 project, click the **XNA Game Studio 3.0** category in the **Project Types** box to gain access to the 3.0 project types. For more information, see [Upgrading XNA Game Studio Projects](#).

Xbox 360 Game (3.1)

XNA Game Studio provides an Xbox 360 Game template that creates and loads a set of starter files. This new project contains basic code that renders a colored background.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can use the Project Designer to modify the values for these properties.

In addition to these properties, assembly information (such as the game title) is stored in the AssemblyInfo.cs file. You can use the [Assembly Information](#) dialog box to modify this information, or you can manually edit the .cs file.

References

References to the following assemblies are added automatically to a new Xbox 360 Game project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Game
- mscorlib
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

Content

The nested content project stores and builds content for the game. For more information, see [Game Content Project](#).

Game1.cs File

This file is a good starting point for adding simple game logic and basic features. It implements a single class (derived from **Game** and called **Game1**), and it overrides five methods: **LoadContent**, **UnloadContent**, **Initialize**, **Draw**, and **Update**. In addition, the **Game1** constructor is defined by this file. Use these methods to initialize your game components, load and render your game content, and handle any input from the user or changes to the game environment.

GameThumbnail.png

The icon appears in the Games Library, and when this game is packed for distribution as a .ccgame. For more information about distribution, see [Sharing Your Game Package](#).

Program.cs File

This file also implements a single class (called **Program**) that provides an entry point to game execution. Usually, little code is added to this file unless the game is fairly advanced.

To create an Xbox 360 game (3.1) project

1. From the **File** menu, click **New Project**.
2. Select the **Xbox 360 Game (3.1)** project type.
3. Type the name for the game project in the **Name** text box.

You can also modify the default values for the **Location** and **Solution Name** controls.

4. Click **OK** to create and load the new project.

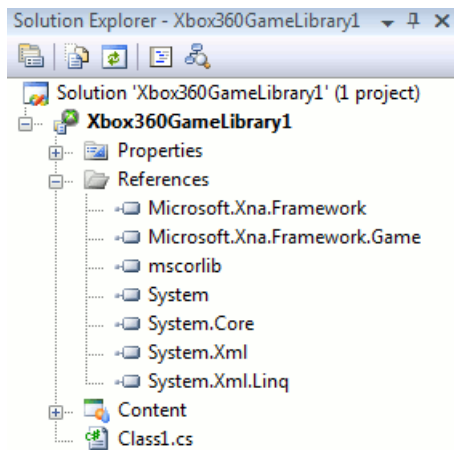
Xbox 360 Game Library (3.1)

XNA Game Studio provides an Xbox 360 Game Library (3.1) template that creates and loads a set of starter files. Typically, projects of this type contain managed classes that implement basic or advanced features used by a game engine or 3D renderer. Once completed, these class libraries can be referenced by other Xbox 360 projects, and they provide common functionality without having that code reside within the game project.

The new project contains basic code that implements an empty library, usable by other Xbox 360 Game projects or Xbox 360 Game libraries.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can use the Project Designer to modify the values for these properties.

In addition to these properties, assembly information (such as the library title) is stored in the AssemblyInfo.cs file. You can use the [Assembly Information](#) dialog box to modify this information, or you can manually edit the .cs file.

References

References to the following assemblies are added automatically to a new Xbox 360 Game Library project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Game
- mscorlib
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

Content

The nested Content project stores and builds content that is to be included in the game library. This is useful for including reusable content assets that support the classes and methods in the game library. For more information, see [Game Content Project](#).

Class1.cs File

This file implements an empty C# class within a namespace. It provides a starting point for the class library.

To create an Xbox 360 Game Library (3.1) project

1. From the **File** menu, click **New Project**.
2. Select the **Xbox 360 Game Library (3.1)** project type.
3. Type the name for the library project in the **Name** text box.

You can also modify the default values for the **Location** and **Solution Name** controls.

4. Click **OK** to create and load the new project.

See Also [Developing Xbox 360 Games](#)
[Sharing Your Game Package](#)

Deploying an Xbox 360 Game

Describes how XNA Game Studio can copy executable and media files to the Xbox 360 retail console once you are ready to deploy them. As you develop your Xbox 360 game, it is convenient to deploy your game executable to your console for testing purposes.

However, before deploying, you need to use the **XNA Game Studio Device Center** dialog box to establish a connection between the console and your computer. For a detailed explanation of this process, see [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#).

Note

For more information about troubleshooting deployment to your console, see [Troubleshooting Xbox 360 Game Deployment](#).

Deploying Your Game

This procedure shows you how to deploy your Xbox 360 game from XNA Game Studio to a console. This assumes you established a connection between that console and your computer.

Note

An XNA Framework game deployed to the Xbox 360 console cannot exceed 2 GB. Also, this means that any single file you deploy must be smaller than 2 GB.

To deploy your Xbox 360 game

1. Load an Xbox 360 Game project into XNA Game Studio.

Make sure the **Output** window is visible so you can follow the progress of your deployment in detail, and so you can tell where a problem occurred, if one does.

2. To display the Output window, from the **View** menu, click **Output**, or press CTRL+W, and then press the letter O key.
3. From the Xbox 360 dashboard, go to the **My Xbox** menu, select **Game Library**, and press the **A** controller button.

Note

Also, you have the option of using the Xbox 360 Guide: From the **Games** page, select **Game Library**, and press the **A** controller button.

4. From the **Game Library** page, select **Community Games** on the **Collections** tab, and press the **A** controller button.
5. Select **XNA Game Studio Connect**, and press the **A** controller button.
6. Select **Launch**, and press the **A** controller button.

The **Waiting for computer connection** page appears.

7. In XNA Game Studio, deploy the game by either right-clicking the solution name and then selecting **Deploy Solution**, or, from the **Build** menu, selecting **Deploy Solution**.

During this process, both the Output window of XNA Game Studio and the **Connect to Computer** screen show the deployment progress. You can view a list of the deployed files and some additional information.

8. Press **B** to return to the **Game Library**.

Your deployed game appears in the list of **Recent Games**.

9. Select the game you just deployed from this menu, and press the **A** controller button.
10. Choose **Play Game**, and press the **A** controller button again to start the game.

Incremental Deployment

Incremental deployment is enabled automatically for XNA Game Studio projects. This means that the IDE tracks which files have been deployed already to your Xbox 360 console. Subsequently, it only deploys updated files. Also, if a file that was part of the previous deployment is no longer in the game, that file is deleted from the console during deployment. (This only applies to previously-deployed files, not to files that were generated on the console.) The output window reflects exactly what actions were taken during deployment.

A full deployment, which deletes any previous game files and copies the entire game to the console, is triggered by the following conditions:

- This is the first time the project has been deployed.
- A different configuration is used than in the last deployment to this device.
- The game was deleted from the Xbox 360 console since the last deployment.
- The solution has been cleaned since the last deployment. (For more information about cleaning the solution, see the "Cleaning the Solution Before You Build" section that follows.)
- The last deployment to the device was from a different source than the currently loaded solution, such as a copy on another Windows-based computer or a .ccgame package.

If you cancel during a deployment, some files may not have transferred. This leaves the game in an indeterminate state. The next incremental deployment will behave as if the last deployment never occurred. It will update the correct files on the Xbox 360 console.

Cleaning the Solution Before You Build

Cleaning your game solution before you build removes all intermediate and output files generated by the build. This forces the next build to regenerate everything.

If you select **Rebuild** on the **Build** menu, the build automatically regenerates everything, but you may also want to clean your solution before you do a build. This is particularly true if you are experiencing unexpected build errors.

In Microsoft Visual C# Express Edition, the **Clean Solution** option is not included in the **Build** menu by default. However, it is easy to add it if you want to make cleaning your solution convenient.

To add the Clean Solution option

1. In XNA Game Studio, right-click the menu or toolbar at the top of the window, and click **Customize**.
2. In the **Customize** dialog box, click the **Build** menu (or any other menu where you want to add the command).
This opens the menu, and leaves it open.
3. From the **Customize** dialog box, click the **Commands** tab, and select **Build** from the **Categories** pane on the left.
4. From the **Commands** pane on the right, drag the **Clean Solution** command into the open **Build** menu, and then drop it where you want it.
5. Close the dialog box.

Now, whenever you want to clean the solution, you can click **Clean Solution** from the menu where you added it.

Managing Several Xbox 360 Consoles with XNA Game Studio Device Center

Using the **XNA Game Studio Device Center** dialog box, XNA Game Studio supports deployment and debugging on multiple Xbox 360s.

The **XNA Game Studio Device Center** does the following:

- Creates connections between your Windows-based computer and each of your Xbox 360 console devices. If you have more than one Xbox 360, you can use **XNA Game Studio Device Center** to create connections between your Windows-based computer and each of your Xbox 360 console devices. Each connection allows you to deploy and debug XNA Framework 3.1 games to a specific Xbox 360 console device.
- Manages the list of available Xbox 360 console devices. This allows you to add and delete devices from the list of devices available for deployment.
- Allows you to set the default Xbox 360 console device to use for deployment. Similar to the default printer in Microsoft Windows, the default Xbox 360 console device is the device to which XNA Game Studio will deploy, unless you specifically select a different Xbox 360 console device for deployment. If there is only one Xbox 360 console device, it is set as the default device.
- Stores the list of Xbox 360 console devices on a per-user basis. Each user has a different list of Xbox 360 console devices to which he or she can deploy, and a different default device to use for deployment.

XNA Game Studio installs the **XNA Game Studio Device Management Toolbar** into Visual Studio. The **XNA Game Studio Device Management Toolbar** does the following:

- When you click the **Launch XNA Game Studio Device Center** button, provides easy access to the **XNA Game Studio Device Center** dialog box.
- Allows you to choose a specific Xbox 360 console device for deployment. Use the **XNA Game Studio Deployment Device** drop-down list. Also, do this if you want to deploy to a different Xbox 360 console device than your default device.

The **XNA Game Studio Device Management Toolbar** automatically appears within Visual Studio when an Xbox 360 Game project or Xbox 360 Library project is loaded in supported versions of Visual Studio tools.

Note

If your startup project is not an Xbox 360 or Zune project, most of the toolbar buttons will be unavailable. To make these buttons available, change your startup project to an Xbox 360 Game or Library project.

For more information about how to add Xbox 360 console devices to the **XNA Game Studio Device Center** dialog box, see [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#). For more information about using the features of **XNA Game Studio Device Center**, see [Using XNA Game Studio Device Center](#).

Managing Multiple Xbox 360 Deployments

As noted previously, **XNA Game Studio Device Center** gives you the option to choose a specific Xbox 360 console device for deployment. This enables you to deploy your game to multiple Xbox 360s. Also, this supports testing and debug of networked games.

If you want to execute the same game instance on all Xbox 360 consoles, you can deploy the same build to all consoles.

However, if you want to debug and modify one instance of the game while the game continues to execute on all other consoles, you can do this by using a unique configuration to build and deploy the instance you want to modify. This is an advanced scenario, but one that is common for debugging networked games.

Deployment Troubleshooting

If you are unable to deploy your Xbox 360 game successfully, see [Troubleshooting Xbox 360 Game Deployment](#).

See Also [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#)
[Using XNA Game Studio Device Center](#)
[Developing Xbox 360 Games](#)

Debugging an Xbox 360 Game

Discusses issues that apply to debugging Xbox 360 games on your retail console.

Note

These issues also apply when debugging a Zune application.

Debugging an Xbox 360 game is similar to the process for [deploying](#) your game, with one exception. Instead of clicking **Deploy Solution**, click **Start Debugging** on the **Debug** menu. To run the game without actually debugging it, click **Start Without Debugging** on the **Debug** menu.

The following features are not available when debugging an Xbox 360 game with XNA Game Studio:

Feature	Description
Interop Debugging	Debugging both managed and native code simultaneously.
Assembly Debugging	Debugging only your assemblies and not system assemblies.
Edit and Continue	Editing the binary content of your game without interrupting your debugging session.
Exception Interrupting	Stopping unhandled exceptions before they unwind, so the user can make changes and retry the operation.
Debugger Visualizers	Displaying a more informative view of a specific data type, such as XML data.
Exception Assistant Support	The additional features provided by Exception Assistant are not available when debugging Xbox 360 games. Information on exceptions is provided by the standard Exceptions dialog box within XNA Game Studio.
Debugging a Running Process	Attaching the debugger to a process that is currently executing.

See Also [Developing Xbox 360 Games](#)

Troubleshooting Xbox 360 Game Deployment

Provides some helpful troubleshooting tips for Xbox 360 deployment if you are unable to establish a successful connection between your computer and your Xbox 360 console.

Be Sure Your Console is Correctly Configured

- You must have an XNA Creators Club membership, which you can purchase from Xbox LIVE Marketplace.
- XNA Game Studio Connect must be installed on the console.
- You must be signed in to Xbox LIVE with an Xbox 360 gamer profile associated with your XNA Creators Club membership.
- XNA Game Studio Connect must be running on the console. This means the connection to Xbox LIVE must remain active. Otherwise, XNA Game Studio Connect closes.

Use a Freshly Generated Key

The key generated by the console in the **XNA Game Studio Connect** page is good only for connecting one computer to one console under one user profile.

A key is valid until a computer connects to the Xbox 360 using that key, or until a new key is generated by the console. For example, if you delete a console from the list of consoles available for deployment, you cannot add it back again using the same key. Instead, you must generate a new key on the console, and use that new key to connect the computer to the console.

Your Connection Keys Might Have Been Modified

If a connection worked once and is now failing, the connection key for the console may have been changed accidentally. If this is the problem, you can fix it simply by adding a new connection, using a new key.

If You Wish to Reset All Connections

You can reset all connections by pushing **Reset All Connections**. This invalidates all connections to the Xbox 360 console device, and generates a new key for a new connection.

Note

If you reset all connections, any Windows-based computer with a connection to this Xbox 360 console device will not be able to connect until it reestablishes a new connection to the Xbox 360 console device.

The Computer and the Xbox Console Must Be on the same Subnet

If the computer and Xbox 360 console are on different subnets, they won't establish a proper connection. Typically, you should connect both machines to the same hub or router. It is best not to use a wireless connection—a weak or noisy signal can cause a failure in the connection and deployment.

To determine whether your computer and console are on the same subnet:

1. On your computer, open a command window (click **Start**, and then click **All Programs**, click **Accessories**, and finally click **Command Prompt**).
2. At the command prompt, type `ipconfig`, and press ENTER.

Make note of both the IP address and of the subnet mask in the configuration listing that appears.

3. On your Xbox 360 dashboard, go to the **My Xbox** menu, and select **System Settings**.

Note

You can use the Xbox 360 Guide to do the same thing: go to the **Settings** blade and select **System Settings**.

4. From **System Settings**, select **Network Settings**, and then click **Configure Network**.

Make note of the IP address and subnet mask.

5. Compare the settings to make sure that the computer and console are on the same subnet.

Check for Firewall Interference

In some cases, your firewall could be blocking communication between your computer and the Xbox 360 console. Verify that the following conditions have been met.

- The xnatrans.exe executable is not blocked on the computer.
- Incoming communication for UDP ports 3825 and 3835 is not blocked.
- Outgoing communication for UDP port 1000 is not blocked.
- Outgoing communication for TCP port 1001 is not blocked.

See [Third-Party Firewall Settings](#) for more details.

See Also [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#)
[Developing Xbox 360 Games](#)
[Using XNA Game Studio Device Center](#)

Developing Zune Games

Describes how to develop games for Zune. Zune game development is similar to Windows game development.

This section discusses how to use the XNA Game Studio environment to develop Zune game titles.

In This Section

[Creating a Zune Game or Library Project](#)

Describes how XNA Game Studio includes a set of project templates that will help you to develop game projects for Zune using the XNA Framework.

[Deploying a Zune Game](#)

Describes how XNA Game Studio can copy executable and media files to a Zune device once you are ready to deploy them.

[Troubleshooting Zune Game Deployment](#)

Provides some helpful troubleshooting tips for Zune deployment if you are unable to establish a successful connection between your computer and your Zune device.

See Also [Zune Programming](#)

Creating a Zune Game or Library Project

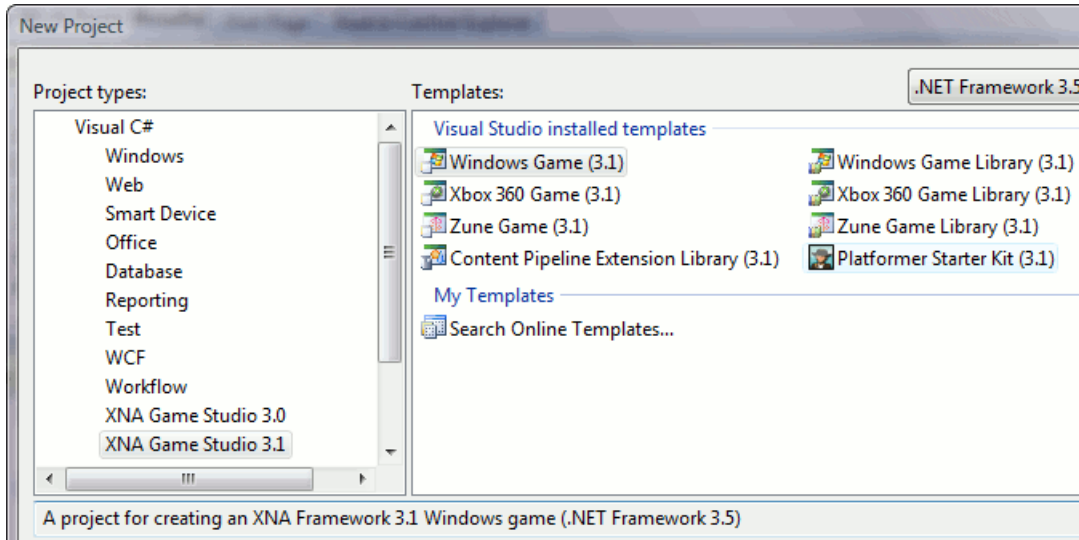
Describes how XNA Game Studio includes a set of project templates that will help you to develop game projects for Zune using the XNA Framework.

Starting a New Project

To start a new project

- To begin a new Zune project, click **File**, and then click **New Project**.

You'll be presented with a dialog that lists a number of project templates.



Project Types

XNA Game Studio provides template types for XNA Framework game development in the **XNA Game Studio 3.1** section of the **Visual C#** project types. The templates offered for developing XNA Framework games for Zune are:

- Zune Game (3.1)—A project for creating an XNA Framework 3.1 game application for Zune.
- Zune Game Library (3.1)—A project for creating an XNA Framework 3.1 game library for Zune.
- Content Pipeline Extension Library (3.1)—A project for creating an XNA Framework 3.1 Content Pipeline Extension Library. For more information, see [Creating a Windows Game or Library Project](#).

For information about XNA Game Studio project templates for Windows or Xbox 360, see [Creating a Windows Game or Library Project](#) and [Creating an Xbox 360 Game or Library Project](#).

Note

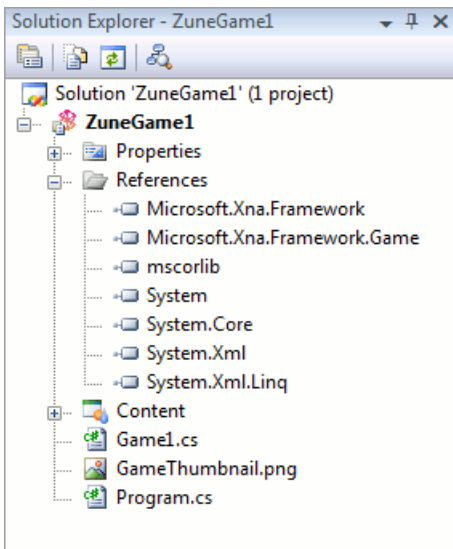
XNA Game Studio 3.1 continues to support creation and use of XNA Game Studio 3.0 projects. If you'd like to create a new XNA Game Studio 3.0 project, click the **XNA Game Studio 3.0** category in the **Project Types** box to gain access to the 3.0 project types. For more information, see [Upgrading XNA Game Studio Projects](#).

Zune Game (3.1)

XNA Game Studio provides an Zune Game template that creates and loads a set of starter files. This new project contains basic code that renders a colored background.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can use the Project Designer to modify the values for these properties.

In addition to these properties, assembly information (such as the game title) is stored in the AssemblyInfo.cs file. You can use the [Assembly Information](#) dialog box to modify this information, or you can manually edit the .cs file.

References

References to the following assemblies are added automatically to a new Zune Game project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Game
- mscorlib
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

Content

The nested content project stores and builds content for the game. For more information, see [Game Content Project](#).

Game1.cs File

This file is a good starting point for adding simple game logic and basic features. It implements a single class (derived from **Game** and called **Game1**), and it overrides five methods: **LoadContent**, **UnloadContent**, **Initialize**, **Draw**, and **Update**. In addition, the **Game1** constructor is defined by this file. Use these methods to initialize your game components, load and render your game content, and handle any input from the user or changes to the game environment.

GameThumbnail.png

The icon appears in the Games Library, and when this game is packed for distribution as a .ccgame. For more information about distribution, see [Sharing Your Game Package](#).

Program.cs File

This file also implements a single class (called **Program**) that provides an entry point to game execution. Usually, little code is added to this file unless the game is fairly advanced.

To create an Zune Game (3.1) project

1. From the **File** menu, click **New Project**.
2. Select the **Zune Game (3.1)** project type.
3. Type the name for the game project in the **Name** text box.

You can also modify the default values for the **Location** and **Solution Name** controls.

4. Click **OK** to create and load the new project.

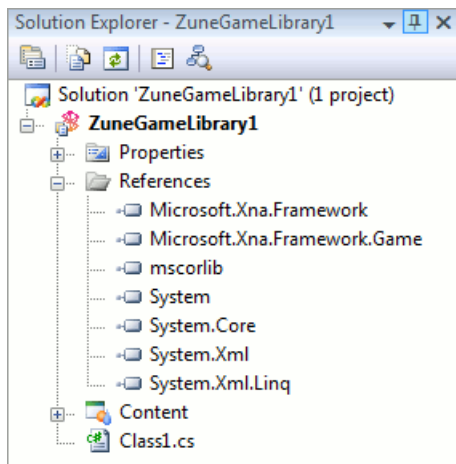
Zune Game Library (3.1)

XNA Game Studio provides an Zune Game Library (3.1) template that creates and loads a set of starter files. Typically, projects of this type contain managed classes that implement basic or advanced features used by a game engine. Once completed, these class libraries can be referenced by other Zune projects. Also, they provide common functionality without having that code reside within the game project.

The new project contains basic code that implements an empty library, usable by other Zune Game projects or Zune Game libraries.

Results

The new starter project contains the following important features.



Project Properties

These properties control many aspects of your current project. Some examples include general application settings, debug settings, and additional project resources. You can use the Project Designer to modify the values for these properties.

In addition to these properties, assembly information (such as the library title) is stored in the AssemblyInfo.cs file. You can use the [Assembly Information](#) dialog box to modify this information, or you can manually edit the .cs file.

References

References to the following assemblies are added automatically to a new Zune Game Library project:

- Microsoft.Xna.Framework
- Microsoft.Xna.Framework.Game
- mscorlib
- System
- System.Core
- System.Xml
- System.Xml.Linq

In addition to these standard assemblies, you can add other assemblies as needed for your project.

Content

The nested Content project stores and builds content that is to be included in the game library. This is useful for including reusable content assets that support the classes and methods in the game library. For more information, see [Game Content Project](#).

Class1.cs File

This file implements an empty C# class within a namespace. It provides a starting point for the class library.

To create an Zune Game Library (3.1) project

1. From the **File** menu, click **New Project**.
2. Select the **Zune Game Library (3.1)** project type.
3. Type the name for the library project in the **Name** text box.

You can also modify the default values for the **Location** and **Solution Name** controls.

4. Click **OK** to create and load the new project.

See Also [Developing Xbox 360 Games](#)
[Creating a Windows Game or Library Project](#)
[Sharing Your Game Package](#)

Deploying a Zune Game

Describes how XNA Game Studio can copy executable and media files to a Zune device once you are ready to deploy them. As you develop your Zune game, it is convenient to be able to deploy your game executable to your device for testing purposes.

However, before deploying, you need to use the **XNA Game Studio Device Center** dialog box to establish a connection between the device and your computer.

Note

For more information about troubleshooting deployment to your Zune device, see [Troubleshooting Zune Game Deployment](#).

Deploying Your Game

This procedure shows you how to deploy your Zune game from XNA Game Studio to a Zune device. This assumes you already have a connection between that device and your computer.

To Deploy Your Zune Game

1. Load a Zune Game project into XNA Game Studio.
2. In XNA Game Studio, deploy the game by either right-clicking the solution name and then selecting **Deploy Solution**, or, from the **Build** menu, selecting **Deploy Solution**.

During this process, both the Output window of XNA Game Studio and the **Connect to Computer** screen show the deployment progress by displaying a list of the deployed files and other information. Also, you can see the names of the files on the device screen as they are deployed.

3. Select the top level **Games** menu on the device to start the game.

Incremental Deployment

Incremental deployment is enabled automatically for XNA Game Studio projects. This means that the IDE tracks which files have already been deployed to your device. Subsequently, it only deploys updated files. Also, if a file that was part of the previous deployment is no longer in the game, that file is deleted from the console during deployment. (This applies only to previously-deployed files, not to files generated on the device.) The output window reflects exactly what actions were taken during deployment.

A full deployment, which deletes any previous game files and copies the entire game to the device, is triggered by the following conditions:

- This is the first time the project has been deployed.
- A different configuration is used than in the last deployment to this device.
- The last deployment to the device was from a different source than the currently loaded solution—a source such as a copy on another Windows-based computer or a .ccgame package.
- The game was deleted from the device since the last deployment.
- The solution has been cleaned since the last deployment. (For more information about cleaning the solution, see the "Cleaning the Solution Before You Build" section that follows.)

Cleaning the Solution Before You Build

Cleaning your game solution before you build removes all intermediate and output files generated by the build. This forces the next build to regenerate everything.

If you select **Rebuild** on the **Build** menu, the build automatically regenerates everything. However, you may also want to clean your solution before doing a build. This is particularly true if you are experiencing unexpected build errors.

In Microsoft Visual C# Express Edition, the **Clean Solution** option is not included in the **Build** menu by default. However, it is easy to add if you want to make cleaning your solution convenient.

To add the Clean Solution option

1. In XNA Game Studio, right-click the menu or toolbar at the top of the window, and then click **Customize**.
2. In the **Customize** dialog box, click on the **Build** menu (or on any other menu where you want to add the command).

This opens the menu, and leaves it open.

3. From the **Customize** dialog box, click the **Commands** tab, and select **Build** from the **Categories** pane on the left.
4. From the **Commands** pane on the right, drag the **Clean Solution** command into the open **Build** menu, and drop it where you want it.
5. Close the dialog box.

Now, whenever you want to clean the solution, you can click **Clean Solution** from the menu where you added it.

Managing Several Zunes with XNA Game Studio Device Center

Using the **XNA Game Studio Device Center** dialog box, XNA Game Studio supports deployment and debugging on multiple Zune devices.

The **XNA Game Studio Device Center** does the following:

- Creates connections between your Windows-based computer and each of your Zune devices. Each connection allows you to deploy and debug XNA Framework 3.1 games to a specific Zune device.
- Manages the list of available Zune devices, allowing you to add and delete devices from the available deployment list.
- Allows you to set the default Zune device to use for deployment. Similar to the default printer in Microsoft Windows, the default Zune is the device to which XNA Game Studio will deploy unless you specifically select a different Zune device for deployment. If there is only one Zune device, it is set as the default device.
- Stores the list of Zune devices on a per-user basis. Each user has a list of different Zune devices to which he or she can select for deployment, and a different default device to use for deployment.

XNA Game Studio installs the **XNA Game Studio Device Management Toolbar** into Visual Studio. The **XNA Game Studio Device Management Toolbar** does the following:

- Gives you easy access to the **XNA Game Studio Device Center** dialog box from within Visual Studio by clicking the **Launch XNA Game Studio Device Center** button.
- Allows you to choose a specific Zune device to deploy by using the **XNA Game Studio Deployment Device** drop-down list. Use this if you want to deploy to a Zune device other than your default device.

The **XNA Game Studio Device Management Toolbar** automatically appears within Visual Studio when an Xbox 360 (or Zune) Game project or Xbox 360 (or Zune) Library project is loaded in supported versions of Visual Studio tools.

Note

If your startup project is not an Xbox 360 or Zune project, most of the toolbar buttons will be unavailable. To make these buttons available, change your startup project to a Zune Game or Library project.

For more information about using the features of **XNA Game Studio Device Center**, see [Using XNA Game Studio Device Center](#).

See Also [Zune Programming](#)
[Debugging an Xbox 360 Game](#)

Troubleshooting Zune Game Deployment

Provides some helpful troubleshooting tips for Zune deployment if you are unable to establish a successful connection between your computer and your Zune device.

When XNA Game Studio Device Center fails to find and add your Zune device, or when XNA Game Studio fails during deployment of a game to your Zune device, you will get the following error message:

```
Could not connect to [Zune name]. Make sure the device is connected, a game is not already running, the screen is not locked, and the Zune software is not running on your computer.
```

One of the following conditions may have caused this failure:

- The Zune device to which you are deploying is not connected to your computer. Make sure the Zune device is securely connected to your computer with the Zune Sync Cable.
- The Zune software currently is running on your computer. You must close the Zune software before deploying to your Zune device.
- A game currently is running on the Zune. Close the game and wait for the Zune device to reboot before you try to deploy again.
- The Zune screen is locked. To deploy to the Zune device, unlock the screen by entering the lock code.

After you correct the condition, wait at least 30 seconds before retrying. If you try to deploy sooner, XNA Game Studio probably won't recognize the change.

See Also [Connecting to your Zune Device with XNA Game Studio](#)

[Developing Zune Games](#)

[Using XNA Game Studio Device Center](#)

[Debugging an Xbox 360 Game](#)

Managing Game Assets

Describes game assets, which are the collection of data files used to support gameplay such as bitmaps, models, textures, or sounds. Game assets are governed by the XNA Framework Content Pipeline.

This section discusses how you can manage the game assets held by a game project.

In This Section

[Game Content Project](#)

Describes a project type (nested within a standard XNA Game Studio project) designed to store all content for a game application.

[Adding Game Content Projects](#)

Describes how XNA Game Studio supports the use of multiple content project folders within a game project.

[Adding Game Assets to Your Game](#)

Demonstrates how to add a texture asset to your game. The same procedure can also be applied to model and sound assets.

[Game Asset Properties](#)

Describes property settings indicating which actions the project system should perform on a game asset file of a supported type.

See Also [Using XNA Game Studio](#)

Game Content Project

Describes a project type (nested within a standard XNA Game Studio project) designed to store all content for a game application.

Overview

A game content project uses the XNA Framework Content Pipeline to build game content (models, textures, sounds, and so forth). You add content items to this project as project items. Item properties define how each content item is imported and processed as part of the build operation. Game content projects are built automatically whenever their parent projects are built. In addition, their configurations and platforms are the same as their parent projects.

Game content projects have a **References** node that lists the assemblies and projects containing the importers and processors currently used in a Content Pipeline build. Individual content items are listed and their properties are accessible using the **Properties** window. You can use either the standard **Add New Item** dialog box or the **Add Existing Item** dialog box to add items to a game content project.

⚠ Caution

The **Build Action** property of any project item moved into the nested content project is set automatically to **Compile**. This value persists even if the same item later is moved out of the nested content project.

Content in Game Libraries

Classes and methods in game libraries may rely on specific content, such as textures and shaders, to perform their game-related functions. For this purpose, game libraries provide their own content subfolders that can contain those assets to ensure they will be available.

When a game project includes reference to a game library, a copy of the output of the game library is included in the output directory of the game project. In this way, both the game library's referenced assembly and its associated content are available at run time.

All content in the game content subproject of a successfully built game library will be compiled through the content pipeline, just as the code of a game library is stored in its compiled form. This permits game projects that reference these reusable elements of the game library to build more quickly.

Content within a game library may be referenced directly by the game project. There is no requirement that the classes or methods of the game library must reference the content in a game library.

Project Designer Properties

These properties are not available to game content projects. All changeable properties are in the Content Pipeline Properties, described below.

Content Pipeline Properties

The content project has additional properties (other than those available in the Project Designer tabs) that specify operating parameters for the XNA Framework Content Pipeline. To access these properties, select the **Content** node in Solution Explorer, then from the **View** menu, select **Properties Window** (or type the function key F4).

Content Root Directory

Specifies the name of the subdirectory that will hold the final output files of pipeline content generated from this project folder. This may be useful if your game project has multiple content projects and you wish to keep the output of each in different subdirectories.

Code-Only Builds

By manually editing a .csproj file, you can build your game project without also building the nested content project. If you are using XNA Game Studio, open the related .csproj file, and add the following property:

```
<SkipNestedContentBuild>true</SkipNestedContentBuild>
```

If you are building from the command-line, use the following command to skip the nested content project:

```
Msbuild /p:SkipNestedContentBuild=true WindowsGame1.csproj
```


For more information on using MSBuild with project files, see [How To: Edit Project Files](#) and [Visual Studio Integration \(MSBuild\)](#).

In addition, you can build only the nested content project from the command line by invoking MSBuild directly on the .contentproj file of your nested content project.

See Also [Adding Game Content Projects](#)

[Managing Game Assets](#)

[Using XNA Game Studio](#)

Adding Game Content Projects

Describes how XNA Game Studio supports the use of multiple content project folders within a game project.

When initially created, an XNA Game Studio game project contains a single subordinate content project to contain game assets to be managed through the content pipeline. It may be convenient to create and maintain additional content projects, perhaps to manage differing types or collections of content.

To add a new content project

1. Open an XNA Framework game or library project in XNA Game Studio.
2. In Solution Explorer, right-click the game project node.
3. Click **Add**, and then click **New Content Project...**
4. In the **Add New Content Project** dialog box, enter the name for the new content project.

The same procedure can also be performed by selecting **Add New Content Project...** from the **Project** menu.

To add an existing content project

1. Open an XNA Framework game or library project in XNA Game Studio.
2. In Solution Explorer, right-click the game project node.
3. Click **Add**, and then click **Existing Content Project...**
4. In the file selection dialog box, navigate to the existing content project file, and then select it.

The same procedure can also be performed by selecting **Add Existing Content Project...** from the **Project** menu.

Note

It is acceptable to add the same content project to projects for different platforms (as in a cross-platform solution). However, if your solution contains multiple projects for the same platform, do not add the same content project to these projects. This will result in unpredictable behavior.

To remove a content project

1. Open an XNA Framework game or library project in XNA Game Studio.
2. In Solution Explorer, expand the game project node and right click the game content project.
3. Click **Remove**.

See Also [Managing Game Assets Using XNA Game Studio](#)

Adding Game Assets to Your Game

Demonstrates how to add a texture asset to your game. The same procedure can also be applied to model and sound assets. It is assumed that an existing Windows, Xbox 360, or Zune game project is loaded in XNA Game Studio.

There are two ways to add a texture asset to your game, either by adding the asset file or by adding a link to the asset.

Tip

Adding an existing asset to your project is quite different from adding an existing item as a link to your project. The first method creates a copy of the asset file and adds the copy to your project. Adding an asset as a link stores only the path to the asset file.

Adding the Texture Asset to the Game Project

When you add a game asset (not as a link), XNA Game Studio makes a copy of the asset file and adds the copy to the content project. For this reason, adding a game asset in this manner may be most appropriate when the asset is used by only one developer in one project and is not expected to be changed.

To add the texture asset to the game project

1. From the Solution Explorer window, right-click the **Content** node, click **Add**, and then click **Existing Item**.
2. Navigate to the location of the texture, and select it.
For this example, the asset is called B1_nebula01.tga.
3. Click the **Add** button.
This creates a copy of the selected asset in your project.
4. Save the solution.

Adding the Texture Asset as a Link to the Game Project

Adding an asset as a link is useful if the referenced asset depends on additional external assets. It ensures that the solution always uses the latest version. For this reason, adding a game asset as a link may be most appropriate when the asset is shared with other people or other game projects, or is likely to be changed.

To add the texture asset as a link to the game project

1. From the Solution Explorer window, right-click the **Content** node, click **Add**, and then click **Existing Item**.
2. Navigate to the location of the texture, and select it.
For this example, the asset is called B1_nebula01.tga.
3. Click the small arrow to the right of the **Add** button, and then click **Add as Link**.
This creates a reference to the selected asset (and not a copy) in your project.
4. Save the solution.

Verifying the Content Importer

You should use this procedure each time you add a game asset. This ensures that the asset will be correctly recognized and processed by the Content Pipeline.

To verify the game asset will use the correct content importer

1. After you add the asset to the solution, open the **Properties** window to verify that you specified the correct importer and processor.
2. To verify, right-click on the file in Solution Explorer, and then, from the context menu, click **Properties**.
For this example, the **Content Importer** is Texture - XNA Framework and the **Content Processor** is Texture - XNA Framework.

For more information on the **Properties** window of a game asset, see [Game Asset Properties](#).

3. Save the solution.

See Also [How To: Load Content](#)

[How To: Draw a Sprite](#)

[Content Pipeline](#)

Game Asset Properties

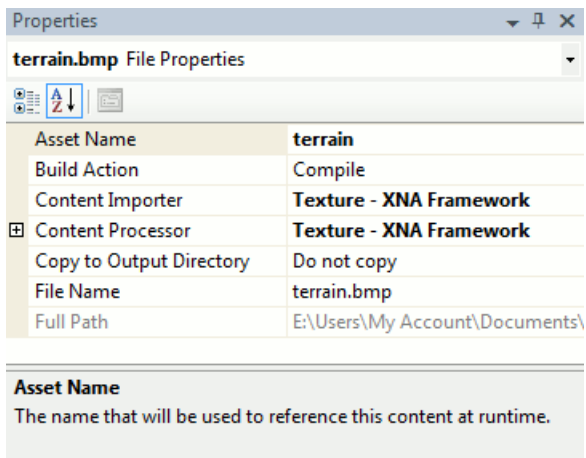
Describes property settings indicating which actions the project system should perform on a game asset file of a supported type.

For example, you can set file properties to indicate the importer and processor to be used for the game asset file.

Note

When adding a game asset of a type not recognized by XNA Game Studio, you need to modify the **Content Processor** and **Content Importer** properties (by specifying a custom processor and custom importer that build assets of this type) before the asset is built by the Content Pipeline.

To examine the properties of a game asset, right-click on the file in Solution Explorer and choose **Properties** on the context menu. The Properties window then appears.



In addition to the [standard properties](#), XNA Game Studio content files have additional custom properties: **Asset Name**, **Content Importer**, and **Content Processor**.

Asset Name

The name of the managed object, generated by the Content Pipeline from the game asset. This name is used at run time to load the managed asset, using the [ContentManager.Load](#) method. The default name is the original file name without the file extension.

Content Importer

The name of the importer for the related game asset. The list contains both custom importers referenced by the content project and standard importers provided by XNA Game Studio.

Content Processor

The name of the processor for the related game asset. The list contains both custom processors referenced by the content project and standard processors provided by XNA Game Studio. Processors that have modifiable parameters are indicated with a small plus sign next to the Content Processor row. Clicking this expands the Content Processor row to display any supported parameters. Modify these parameter values as you would other properties in this pane. For more information on parameterized processors, see [Parameterized Processors](#).

See Also [Overview of the Content Pipeline](#)
[Using a Custom Importer or Content Processor](#)
[Using XNA Game Studio](#)

Developing Cross-Platform Games

Describes how XNA Game Studio and the XNA Framework facilitate the development of games that work on several platforms.

This section discusses how you can use the features of the XNA Game Studio to develop cross-platform games.

In This Section

[Creating Cross-Platform Games](#)

Describes how Game Studio facilitates developing a game that will run on multiple platforms.

[Cross-Platform Game Project Converter](#)

Describes how XNA Game Studio facilitates the process of converting a game for Windows, Xbox 360, or Zune to any of the other two game platforms.

[Platform-Specific Content Projects](#)

Describes how XNA Game Studio supports maintaining platform-specific game assets through the use of multiple content projects within a game project.

[Managing Cross-Platform Builds and Deployment](#)

Describes how to build and deploy individual projects within a cross-platform solution.

[Cross-Platform Conditional Compilation Symbols](#)

Describes the conditional compilation variables available for maintaining common source code in XNA Game Studio cross-platform projects.

[How To: Create a Cross-Platform Game Solution](#)

Demonstrates how to create a cross-platform solution and manage game assets between platform projects.

See Also [Using XNA Game Studio](#)

Creating Cross-Platform Games

Describes how Game Studio facilitates developing a game that will run on multiple platforms.

The XNA Framework makes developing a game with rich play on any Microsoft platform easy. Even better, once your game works on one platform, it can run on any other platform supported by the XNA Framework, using the same base source code with minimal changes.

For example, the same XNA Framework game could be built three ways: the first to be played on a desktop computer running Windows, the second to be downloaded to an Xbox 360, and a third to load onto a Zune device.

XNA Game Studio provides several features that facilitate development and management of games designed to run on multiple platforms.

The Structure of a Cross-Platform Game in Game Studio

When you develop a game to run on multiple platforms, there must be a separate project for each platform. Each project may reference the same source code files and game asset files, but building each project will create an executable that can run on its designated platform.

Method 1: A Solution File for Each Platform.

One approach to maintaining separate projects for each platform might be to create a solution file for each supported platform, each with its own single project. If the projects in each solution reference the same source and game asset files, changes to those files will be reflected in each solution.

The drawback to this method is that each solution will have to be maintained separately. When you make changes to a project, such as adding new source and game asset files, renaming or removing items, the changes you make in one solution will not be reflected automatically in other solutions.

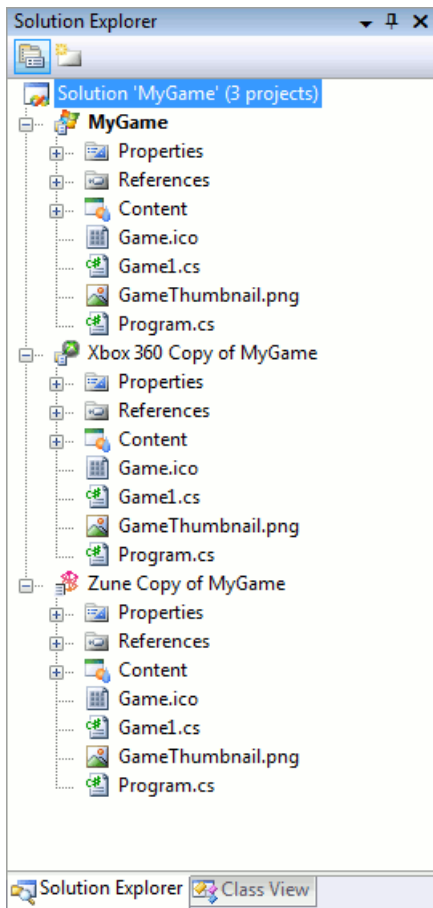
Method 2: A Solution File with Multiple Projects

A cross-platform game is best maintained as multiple projects within a single solution, so that Game Studio can more easily synchronize changes across all platforms.

For example, the Game Studio solution for MyGame might house a project folder for the following:

1. Windows version of MyGame.
2. Xbox 360 version of MyGame.
3. Zune version of MyGame.

Each of these projects will reference the same C# code files and game asset file, but each will build a version that can execute on the intended platform.



When you use the [Cross-Platform Conversion Wizards](#) to create these parallel projects, the project created is [linked to the original project](#).

Managing Platform-Specific Content

Source code and game asset files should be shared for all platforms as much as possible. However, some files may need to have unique versions for different platforms.

This requirement occurs most frequently for game assets in the content pipeline. For example, to produce the highest quality images, graphic files for a game designed to run on the Xbox 360 platform may be very high resolution bitmaps. The Zune platform, however, has a much smaller screen and supports much lower-resolution graphics. For this reason, your game projects are likely to reference differing game assets that are appropriate to each platform – large, high-resolution graphics for the Xbox 360 and small, low-resolution graphics for the Zune.

At the same time, your game may still have content assets, such as menus, that are shared across all platforms. For these assets, it will still be useful for each project to maintain a content project folder that is synchronized across all platform projects.

To accommodate this, Game Studio enables a project to contain multiple content projects. The first project, synchronized to the content folders of all other projects, can contain the shared assets. Subsequent content project folders may contain game assets specific to that platform, and can be made to be exclusive to that parent project (not synchronized to other projects).

Unsynchronized content projects are created by adding a content project, then removing its synchronized versions in other game projects. You can use the [Add Game Content Project](#) procedure to add new platform-specific content projects to code projects.

The topic [How To: Create a Cross-Platform Game Solution](#) demonstrates how this can be used to manage platform-specific content across multiple platforms.

You can create several content project folders within a game project. You are not restricted to the two described in our example. If you find it convenient to maintain more, Game Studio will permit it.

See Also [Developing Cross-Platform Games](#)

Cross-Platform Game Project Converter

Describes how XNA Game Studio facilitates the process of converting a game for Windows, Xbox 360, or Zune to any of the other two game platforms.

The conversion process creates a new game project of the appropriate type. The new game project will be created in the same folder as the source project, and will share content and settings from the original game project.

- [Limitations of Game Conversion](#)
- [Cross-Platform Game Conversion](#)
- [Changes Made to a Project During Conversion](#)
- [Linked Projects After Conversion](#)
- [References the Conversion Utility Cannot Replace](#)
- [Changes that May Be Necessary after Conversion](#)

Limitations of Game Conversion

The conversion utility is not meant to be an end-to-end conversion, but it greatly simplifies the process of converting a game to the corresponding platform. Specific code changes will still be required. For example, if a Windows game is controlled by the keyboard and mouse, the converted game will not be changed to respond to an Xbox 360 game controller. You can use the [cross-platform conditional compilation symbols](#) to create sections of code that will execute only on the appropriate platform.

Cross-Platform Game Conversion

To convert a game for another platform

1. Load the game into a supported version of Visual Studio tools.
2. From the **Project** menu, choose the **Create Copy of [project name]** item, where the [project name] will be the name of the game project.

The presented items in the **Project** menu will vary according to the designated platform of the currently selected project:

Windows	Xbox 360	Zune
Create Copy of [Project Name] for Xbox 360	Create Copy of [Project Name] for Windows	Create Copy of [Project Name] for Windows
Create Copy of [Project Name] for Zune	Create Copy of [Project Name] for Zune	Create Copy of [Project Name] for Xbox 360

When complete, a new version of the game project will be added to your solution. It will be named for the platform (for example, "Xbox 360 Copy of [Project Name]" or "Windows Copy of [Project Name]"). The new game project will be selected.

3. Press F2 if you want to rename the project from the name given by the conversion utility.

Note

The solution may only contain one copy of a project for each platform. Solutions that do not conform to this will be unable to correctly synchronize projects and produce an error message during conversion.

Changes Made to a Project During Conversion

- All references to XNA Framework assemblies will be replaced by their equivalents for the other platform.
- All original configurations will be replaced by default configurations for the new platform.
- The XnaPlatform project property will change to that of the new platform.
- The [Content Build Compression](#) properties are set to the default values for the new platform.
- The ProjectGuid property will be replaced. This is not the same as the GUID assembly attribute (usually found in AssemblyInfo.cs), which will not be changed.
- The default Platform property and project output type settings will be replaced with values for the appropriate target platform.

Linked Projects After Conversion

The new project created by the conversion wizard shares code, content, and settings from the original game project by being linked.

Projects copied through the conversion wizard are linked so that:

- An item or folder added to any project is also added to the other projects.
- An item or folder renamed in any project is also renamed in the other projects.
- An item or folder deleted from any project is also deleted from the other projects.

All other actions not specifically cited above are not synchronized between projects. Of particular note, this includes the following actions:

- Excluding an item from a project has no effect on other projects. This is useful for items that are inappropriate for one or more platforms, yet needed for another platform (or platforms). For an example, see [How To: Create a Cross-Platform Game Solution](#).
- Changing properties in a project or folder has no effect on other projects.

The exception to this behavior is items in and properties of linked content project folders (for example, the **Content** project or content projects that you [create](#)) that exist within game projects. Excluding an item from a content project always excludes the item from all of its linked content projects. (As examining the properties of the linked content project folders will show, each is referencing the same project file. So, adding or removing an item to one will affect all.) This behavior is unique to content projects.

Note

Folders that are empty (as when they are first created) are not synchronized between projects. Once items are placed into a folder (so that it is no longer empty), the folder is synchronized across all other projects.

References the Conversion Utility Cannot Replace

- HintPath metadata (added using browse-to-file) and strong name references will be changed into simple-named references.
- Project references to non-XNA Framework projects will be deleted.

Note

Unlike references to non-XNA Framework projects, all referenced XNA Framework game or game library projects are recursively converted. This means that if an XNA Framework project references another in the solution (with a project-to-project reference), conversion of the referencing project automatically converts the referenced project. The result is the converted copy of the first project referencing the converted copy of the second project.

For example, if you are converting a Windows game to the Xbox 360 that references a Windows game library in the same solution, conversion also occurs for the Windows game library. After a completed conversion, the result is an Xbox 360 game (a converted copy of the Windows game) that references an Xbox 360 game library (a converted copy of the Windows game library).

- COM references will be deleted. The conversion wizard will provide a warning to the output window when this occurs.
- COM file references will be deleted. The conversion wizard will provide a warning to the output window when this occurs.
- Web references are removed, possibly resulting in compile time errors for the converted project. For example, any code that instantiates a Web reference-related class or invokes its methods will cause compile-time errors. The conversion wizard will provide a warning to the output window when this occurs.

Caution

If you need to remove a file from only one project, use the **Exclude from Project** command rather than the **Delete** command. The **Delete** command removes the file from disk, making it inaccessible to the other project.

Changes that May Be Necessary after Conversion

- In some cases, a converted reference (for example, a strong name reference) may fail to resolve. If this happens, either delete the reference or replace it with a reference to an assembly version for the correct platform.

For example, if a reference to System.Data (often added automatically to Windows projects) was converted for the Xbox 360, you must remove it because System.Data are not supported on Xbox 360. However, if the converted reference was to something like a third-party game engine, you might be able to change the reference to an Xbox 360 version of the engine instead.

If a converted reference is deleted, ensure that any platform-specific code does not depend on that reference.

- Any code specific to one target platform, for example, mouse-based user input in a Windows game, will not be converted for the other platform. You must manually change this code to use an Xbox 360-compatible input device.

Other common examples include any .settings files (due to missing types), code generated from any Windows Forms in the project, and certain types of resources. For example, string or binary resources are fine but bitmap resources will fail.

See Also [Developing Cross-Platform Games](#)
[Creating a Windows Game or Library Project](#)
[Creating an Xbox 360 Game or Library Project](#)
[Creating a Zune Game or Library Project](#)

Platform-Specific Content Projects

Describes how XNA Game Studio supports maintaining platform-specific game assets through the use of multiple content projects within a game project.

A cross-platform solution in XNA Game Studio is composed of multiple game projects, with one project for each platform. The projects are synchronized with each other such that adding or renaming a file in one project is reflected automatically in the other projects.

Content projects that are [added to a game project](#) follow the same synchronization rules. This means that adding a content project to one game project will automatically add the same content project to all other game projects in the XNA Game Studio solution. For each of the other projects, all references to XNA Framework assemblies will be appropriate for the project's platform.

It may be useful to maintain a content project that is exclusive to a game project, such as for game assets that are specifically meant for use on that project's platform. For example, you may want to use high-resolution bitmaps for a project that will run on the Windows platform, but use low-resolution versions for the Zune project.

To create content projects that are not synchronized with other game projects, you must first add a content project to the needed game project, and then remove the synchronized projects in the game projects where they are not needed.

To create an exclusive content project for a Windows game

This example procedure assumes that the XNA Game Studio solution contains three game projects: for Windows, Xbox 360, and Zune.

1. In Solution Explorer, right-click the Windows game project node (for example, MyGame).
2. Click **Add**, and then click **New Content Project...**
3. In the **Add New Content Project** dialog box, enter the name for the new content project (for example, MyWindowsContent).
4. In Solution Explorer, expand the Xbox 360 game project node (for example, Xbox 360 Copy of MyGame).
5. Within the Xbox 360 game project, right-click the new content project (MyWindowsContent), and select **Remove**.
6. In Solution Explorer, expand the Zune game project node (for example, Zune Copy of MyGame).
7. Within the Zune game project, right-click the new content project (MyWindowsContent) and select **Remove**.

See Also [Adding Game Content Projects](#)
[Developing Cross-Platform Games](#)

Managing Cross-Platform Builds and Deployment

Describes how to build and deploy individual projects within a cross-platform solution.

When you are developing a cross-platform game in XNA Game Studio, your solution will be composed of a project for each possible platform. You need to understand how to use the features within Visual Studio that control the particular projects that are built and deployed following the **Start** (F5) key and **Build** commands.

For example, if you are developing a cross-platform project for both Windows and Xbox 360, you will want to limit game build and deployment solely to the Xbox 360 project when you are trying to develop and debug the Xbox 360 version of the game.

You will need to know how to do the following:

1. Set the deployment options to limit builds and deployment to the startup project.
2. Set one of the projects in your cross-platform solution as the startup project.

You can also use the Solution Configuration to control the characteristics of the build.

Setting Deployment Options

You can use the **Build and Run** settings to limit **Start** and **Build** commands to the startup project.

To set Build and Run options

1. From the **Tools** menu, click **Options...**
2. If you are using Visual C# Express, make sure the check box in the **Options** dialog box at the bottom left-hand corner labeled **Show all settings** is checked.
3. In the **Options** dialog box, expand the **Projects and Solutions** node, and select the **Build and Run** options set.
4. Check the box labeled **Only build startup projects and dependencies on Run** (it is unchecked by default).

This setting will limit build and deployment to the startup project.

In addition, you may want to set the option that automatically sets the currently selected project as the startup project.

To automatically set the startup project

- Check the box labeled **For new solutions use the currently selected project as the startup project** (it is unchecked by default).

This setting will establish that the currently selected project is the startup project. This option will only take effect for new solutions.

Setting the Startup Project Manually

The startup project is the project (or projects) that will be run by Visual Studio when you start the debugger. If you set the **Build and Run** settings as described previously, the **Start** command (also initiated by the **F5** key) will be limited solely to the startup project (or projects).

Further, if the startup project is set automatically to the currently selected project, it is not necessary to manually set the startup project.

The startup project name always appears in boldface in Solution Explorer.

To set the startup project manually

1. In Solution Explorer, right-click the project you want to set as the startup project.
2. In the context menu, click **Set as StartUp Project**.

Also, you can use Solution Explorer to select the project or, from the **Project** menu, select the project by clicking **Set as StartUp Project**.

Using Solution Configurations

Solution configurations store solution level properties that direct which style of object and executable output are produced by the **Start** (F5) key and **Build** commands. In a cross-platform solution in XNA Game Studio, the solution configuration platform setting is a composite of the platform settings for each project in the solution.

The **Solution Configuration** control in the Visual Studio toolbar establishes the "Active" (or default) configuration for all projects in the solution. For example, the active configuration may be set to the "Debug" configuration, which includes all symbol information and minimal compiler optimization; or it may be set to the "Release" configuration. Changing this setting will affect the build of all projects set to the active configuration.

The **Solution Platforms** control in the Visual Studio toolbar establishes which platform version of the projects in a solution will be built and deployed. In a cross-platform solution, the options may be "Mixed Platforms" (that is, all that are specified in the solution configuration), "x86" (for Windows), "Xbox 360," or "Zune."

For example, if you set the solution platform to "Zune," only the Zune projects in the solution will be built and deployed. Xbox 360 or Windows projects are not affected.

Note that this differs from the startup project, which determines which project is run, but not which project is built and deployed. If the Windows project is set as the startup project, but the solution platform is set to "Mixed Platforms," all projects will be built when you press the F5 key, but only the Windows project will execute. Using the previous example, if you change the solution platform setting to "Zune," only the Zune project will be built, but it won't run. The Windows project will not be built, and Visual Studio will run the last built (and perhaps out of date) Windows executable.

To limit Visual Studio to both building and running only the platform version you are currently working on, the solution platform setting should match the platform of the startup project.

You can also limit action to a specific project by right-clicking the project in **Solution Explorer**, and selecting the desired action (such as **Build**, **Deploy** or **Debug**) that appears in the context menu.

Each project may have its own configuration settings, which may override the active configuration. The project's configurations also will specify the target platform (for example, Windows, Xbox 360, or Zune). The platform settings are set automatically for a project when you use the **Create Copy...** command to create the synchronized project for a new platform.

To change the project configuration settings

- In Solution Explorer, right-click the solution, and click **Configuration Manager** on the context menu.

Note

You can also start the Configuration Manager by selecting it in the drop-down **Solution Configurations** menu on the toolbar.

See Also [Adding Game Content Projects](#)
[Developing Cross-Platform Games](#)

Cross-Platform Conditional Compilation Symbols

Describes the conditional compilation variables available for maintaining common source code in XNA Game Studio cross-platform projects.

XNA Game Studio projects define symbols for each compatible platform for use with conditional compilation directives such as **#if**. You can use this to maintain source code that acts differently on Windows, Xbox 360 or Zune when that source is shared between projects.

The conditional compilation symbols for each compatible platform are:

Windows Platform	Xbox 360 Platform	Zune Platform
WINDOWS	XBOX or XBOX360	ZUNE

Usage

Where source code needs to support different actions when executing on different platforms, it should use the XNA Game Studio conditional compilation symbols with the **#if**, **#else** and **#elif** directives.

For example, a game that needs to execute on Windows, Xbox 360, and Zune, but calls different services (such as for input devices) for each platform might use the directives in a style similar to this:

```
#if WINDOWS

// Execute code that is specific to Windows

#elif XBOX

// Execute code that is specific to Xbox 360

#elif ZUNE

// Execute code that is specific to Zune

#else

// Print a compile-time error message
#error The platform is not specified or is unsupported by this game.

#endif
```

[How To: Draw a Sprite](#) shows an example of conditional compilation where the available screen space is calculated differently for Xbox 360 and Windows.

Definition

The **Conditional compilation symbols** are defined in the **Build** tab of the Project Designer properties for each project. These symbols are properly established when the XNA Game Studio project is first created or converted for another platform through the [conversion wizard](#).

See Also [Zune Programming Considerations](#)

[Xbox 360 Programming Considerations](#)

[Developing Cross-Platform Games](#)

[EffectProcessor.Process](#)

How To: Create a Cross-Platform Game Solution

Demonstrates how to create a cross-platform solution and manage game assets between platform projects.

This tutorial shows how create a solution with multiple synchronized projects where each project holds the same base source code, but the object generated from each is targeted to execute on a different XNA Game Studio-compatible platform. It also shows how to create game content projects within each platform project that hold non-synchronized or partially synchronized content specific to each platform.

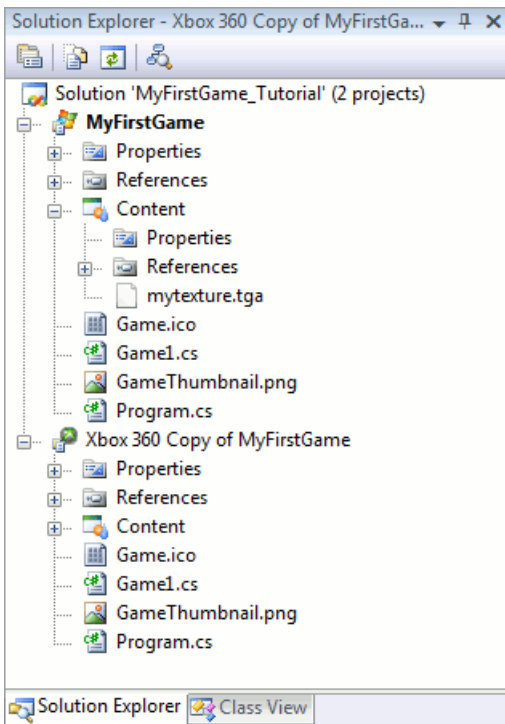
The scope of the tutorial is limited to demonstrating how to use the synchronized project features of Visual Studio to manage a cross-platform solution. It does not demonstrate changes to the source code that would likely be needed in order to execute effectively on different platforms.

Creating a Cross-Platform Solution

To create a game solution with multiple projects for Windows, Xbox 360 and Zune

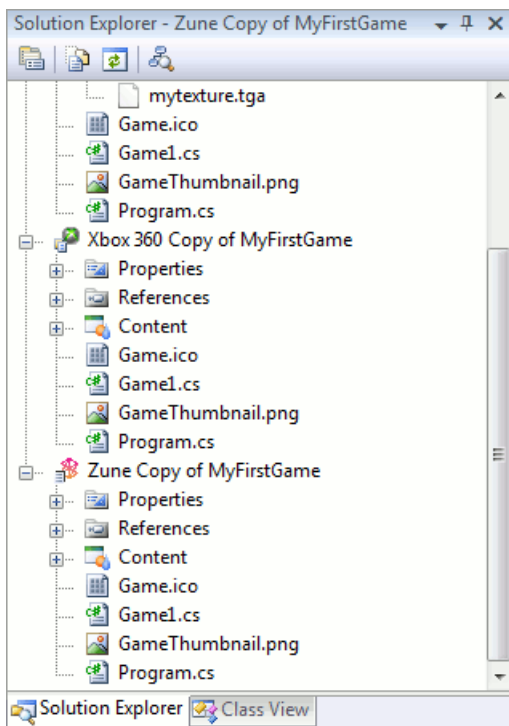
1. Follow the steps of [Your First Game: Microsoft XNA Game Studio in 2D](#) to create a Windows game for XNA Game Studio.
2. In Solution Explorer, select the **MyFirstGame** project.
3. From the **Project** menu, click **Create Copy of MyFirstGame for Xbox 360**.

When complete, a new version of the game project named "Xbox 360 Copy of MyFirstGame" will be added to your solution.



4. In Solution Explorer, select the **MyFirstGame** project once again.
5. From the **Project** menu, click **Create Copy of MyFirstGame for Zune**.

When complete, a new version of the game project named "Zune Copy of MyFirstGame" will be added to your solution.



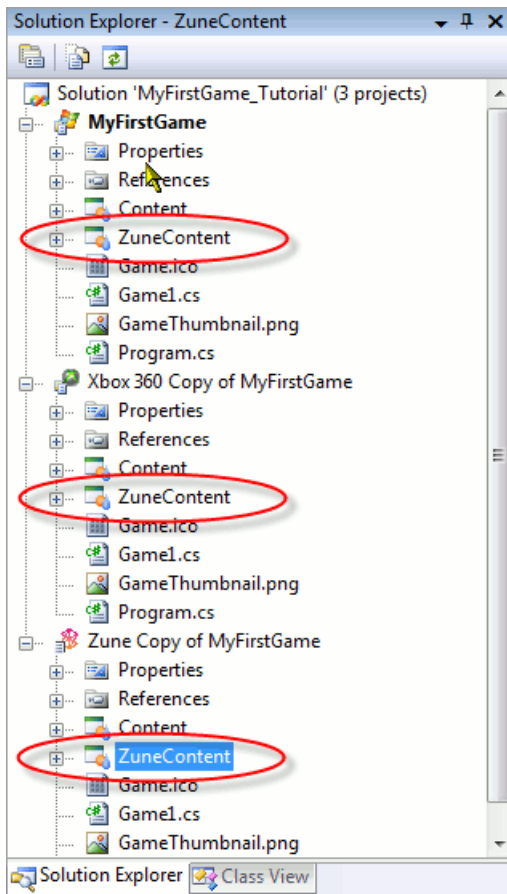
Creating Platform-Specific Content

Because different devices have different characteristics, you may need to maintain different content that is specific to each platform. In the example below, we create a content project for the Zune project that will contain low-resolution textures compatible with that device. We will also create content projects that are synchronized between the Windows project and the Xbox 360 project. These projects hold high-resolution textures that are compatible with both platforms, but are not part of the Zune project.

To create content projects exclusive to each platform project

1. In Solution Explorer, right-click the **Zune Copy of MyFirstGame** project, select **Add**, and then click **New Content Project...**
2. In the **Add New Content Project** dialog box, enter "ZuneContent" as the name for the new content project, and then click **Add**.

A new "ZuneContent" project has been added to all three projects.



3. Right-click the **ZuneContent** project in **MyFirstGame**, and then click **Remove**.

4. Right-click the **ZuneContent** project in **Xbox 360 Copy of MyFirstGame**, and then click **Remove**.

At this point, a "ZuneContent" project exists only in "Zune Copy of MyFirstGame." Any items added to this content project will be used exclusively by the Zune version of the game.

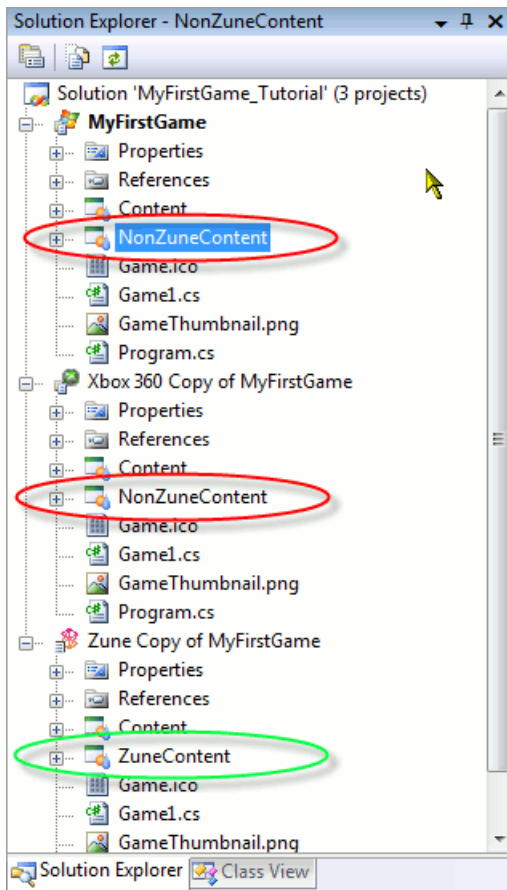
5. In Solution Explorer, right-click the **MyFirstGame** project, select **Add**, and then click **New Content Project...**

6. In the **Add New Content Project** dialog box, enter "NonZuneContent" as the name for the new content project, and then click **Add**.

A new "NonZuneContent" project has been added to all three projects.

7. Right-click the **NonZuneContent** project in **Zune Copy of MyFirstGame**, and click **Remove**.

The solution now has "NonZuneContent" projects in the Windows and Xbox 360 projects, and a "ZuneContent" project in the Zune project.



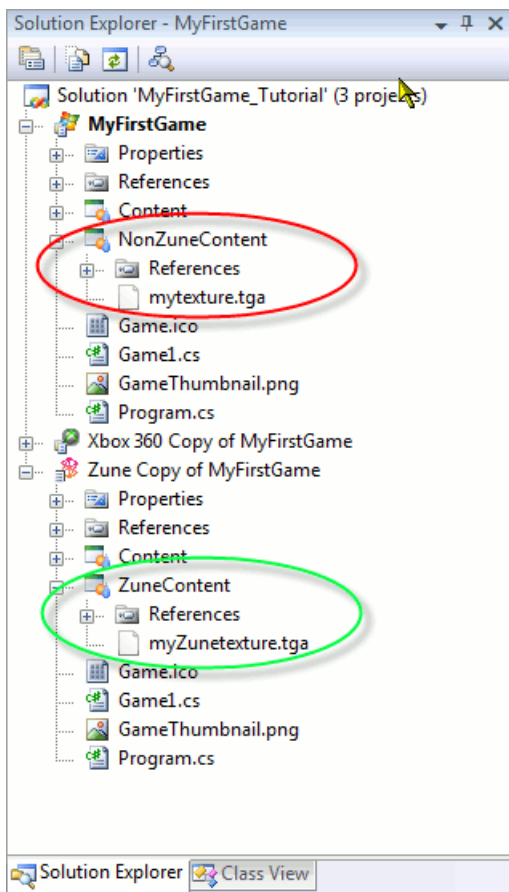
To populate the platform-specific content projects

1. In Solution Explorer, in the **MyFirstGame** project, click and drag the bitmap asset (for example, mytexture.tga) from the **Content** project to the **NonZuneContent** project.
2. In the **Content** project, right-click the bitmap asset (mytexture.tga), and select **Exclude From Project**.

The bitmap asset should now be in the "NonZuneContent" projects of both "MyFirstGame" and "Xbox 360 Copy of My First Game," and no longer in any of the projects' "Content" projects.

3. In Solution Explorer, in the **Zune Copy of MyFirstGame** project, right-click the **ZuneContent** project, select **Add**, and then click **Existing Item**.
4. In the **Add Existing Item** dialog box, select a bitmap file to add (for example, myZunetexture.tga).

This bitmap should be of appropriate resolution for the Zune device. You should now have the bitmap asset common to the Windows and Xbox 360 projects (mytexture.tga) in "NonZuneContent" and the bitmap asset for Zune (myZunetexture.tga) in "ZuneContent".



5. Right-click the newly-added file (myZunetexture.tga), and click **Properties**.
6. In the Properties pane, change the **AssetName** property to match the same name as the **AssetName** property of the version in the "NonZuneContent" project (for example, mytexture).

Setting this property to the same value will help the common source code to load the correct asset for the project's platform.

See Also [Developing Cross-Platform Games](#)

Game Studio Features

Describes how XNA Game Studio provides numerous feature extensions to Visual Studio and Visual C# Express to help create eye-popping games.

This section discusses the special features of the XNA Game Studio development environment.

In This Section

[Using XNA Game Studio Device Center](#)

Describes how to use the XNA Game Studio Device Center.

[Visual Studio Unsupported Features](#)

Describes the features of Visual Studio that are not supported for XNA Game Studio.

[Visual Studio Differences Between Game Platforms](#)

Describes the differences in the appearance and behavior of the integrated development environment (IDE) of XNA Game Studio when developing for the Xbox 360 or Zune.

[Project Properties](#)

Describes the unique property settings of Project Designer in XNA Game Studio.

[Game Component Development](#)

Describes how to use XNA Game Studio to develop custom game components for your game projects.

[Extending Game Studio](#)

Describes how XNA Game Studio can be customized to support special functions to aid development of games.

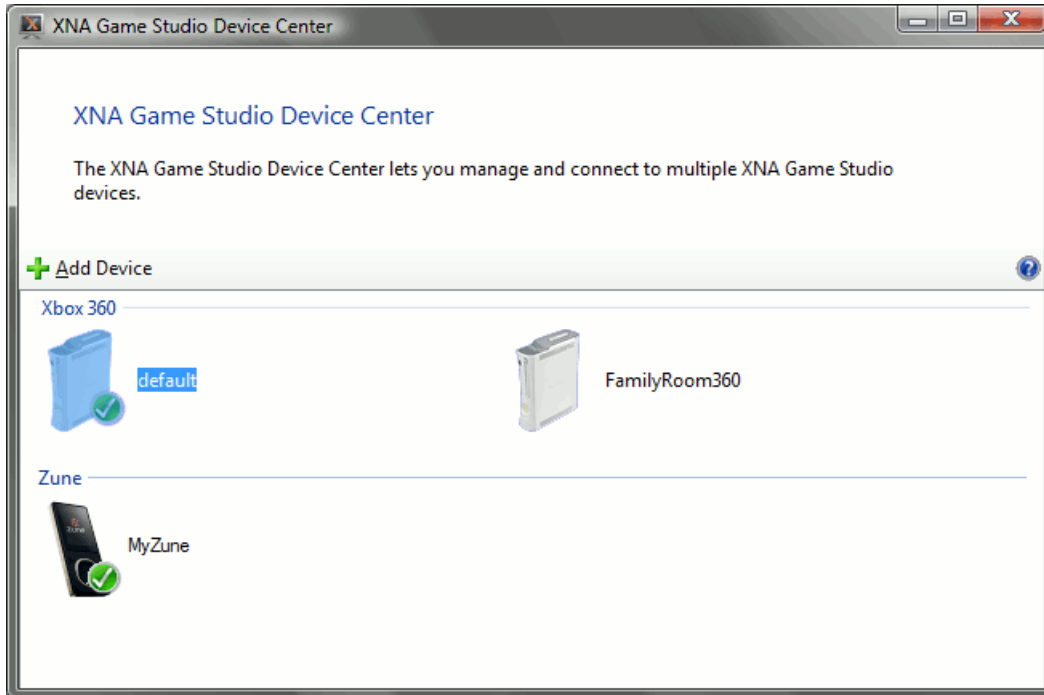
See Also [Using XNA Game Studio](#)

Using XNA Game Studio Device Center

Describes how to use the XNA Game Studio Device Center. The XNA Game Studio Device Center enables you make one or more game devices available for deployment of your games. These can be either Xbox 360 consoles or Zune devices. XNA Game Studio Device Center also lets you set the default device for deployment.

Accessing the XNA Game Studio Device Center

To access the XNA Game Studio Device Center, you can click the **Launch XNA Game Studio Device Center** on the **XNA Game Studio Device Management toolbar** from within Visual Studio.



Another way to access the XNA Game Studio Device Center is to use the **Start** menu.

- From the **Start** menu, choose **Programs**, choose **XNA Game Studio 3.1**, and then click **XNA Game Studio Device Center**.

When XNA Game Studio Device Center appears, it shows a list of game devices available for deploying and debugging XNA Game Studio games.

Adding a Game Device to the List of Devices Available for Deployment

You can use the **XNA Game Studio Device Center** to add a game device (either an Xbox 360 console or a Zune).

To add a game device

- Perform one of the following steps:
 - On the left-hand side of the screen, click **Add Device**.
 - or-
 - From the Visual Studio **XNA Game Studio Device Management** toolbar, click **Add a New Device**.

Either action brings up the **XNA Game Studio Devices** dialog box. This enables you to create a connection between your Windows-based computer and your game device.

You can find detailed step-by-step instructions on how to create a connection and add a game device to the list of devices in the following documents:

- [Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#)
- [Connecting to your Zune Device with XNA Game Studio](#)

Setting the Default Xbox 360 Console for Deployment

The default Xbox 360 is the console automatically selected for deployment and debugging when developing Xbox 360 console games with XNA Game Studio. The current default device is shared by all instances of XNA Game Studio that are running.

To set an Xbox 360 console as the default Xbox 360 device, right-click the icon for the device, and then click **Set as Default Xbox 360**.

If you have only one Xbox 360 console, that console automatically is selected as the default device.

Setting the Default Zune for Deployment

The default Zune is the Zune device automatically selected for deployment and debugging when developing Zune games with XNA Game Studio. The current default device is shared by all instances of XNA Game Studio that are running.

To set a Zune device as the default, right-click the icon for the device, and then click **Set as Default Zune**.

If you have only one Zune, that device automatically is selected as the default device.

Taking a Screen Capture from an Xbox 360 Console or Zune

The XNA Game Studio device center provides the ability to take a screen capture of an executing game on an Xbox 360 console or Zune. To take a screen capture, the following conditions must be met:

- The Xbox 360 console or Zune must be properly connected to the PC.
- The Xbox 360 console or Zune must be executing a game that was launched from Visual Studio.
- The game executing must have been built using the Debug configuration.

To take a screen capture through the Device Center

Once your game is deployed and executing on the game device according to the above conditions:

1. Right-click the icon for the device.
2. From the context menu, select **Take a Screen Capture**.

The captured screen will be displayed, and you can save it as a graphics file.

Removing a Game Device from the List of Devices

To remove a game device from the list of devices, either right-click the icon for the device and click **Remove**, or press the Delete key.

Using the List of Devices

The list of devices in XNA Game Studio Device Center is unique to each user. Each user can have a different list of game devices, and a different default device.

Troubleshooting Your Connection

If you are unable to establish a successful connection between your computer and your game, see the troubleshooting guide:

- [Troubleshooting Xbox 360 Game Deployment](#)
- [Troubleshooting Zune Game Deployment](#)

See Also

General Development Topics

[Game Studio Features](#)

Xbox 360 Development

[Connecting to Your Xbox 360 Console with XNA Game Studio 3.1](#)

[Troubleshooting Xbox 360 Game Deployment](#)

[Developing Xbox 360 Games](#)

Zune Development

[Connecting to your Zune Device with XNA Game Studio](#)

[Troubleshooting Zune Game Deployment](#)

[Developing Zune Games](#)

Visual Studio Unsupported Features

Describes the features of Visual Studio that are not supported for XNA Game Studio.

The following Microsoft Visual Studio items are not supported for XNA Game Studio projects:

- Creating or running Visual Studio Team System 2008 Test Edition unit tests for an Xbox 360 or Zune Game project. For this reason, the **Create Unit Tests...** command is hidden in the **Code Editor** context menu.
- Adding Windows Form, User Control, or Component classes to an Xbox 360 or Zune Game project. For this reason, these options are not available from the **Project** context menu.
- Adding service references to an XNA Game Studio project. For this reason, the **Add Service Reference** command is not available from the **References** context menu or the **Project** context menu.
- Adding Web references to an XNA Game Studio project. For this reason, the **Add Web Reference** command is not available from the **References** context menu or the **Project** context menu.
- Adding data sources. Data sources are not supported on the Xbox 360 or Zune platforms. For this reason, the **Data** menu is hidden when an Xbox 360 or Zune Game project is active.
- Publishing the project using the Visual Studio **Publish** command. This command is available from the **Build** menu for Windows projects only, as described in the topic [Distributing Your Finished Windows Game](#). This command is not available for projects on the Xbox 360 or Zune platforms. You should use the **Deploy** command instead. For more information on deployment, see [Deploying an Xbox 360 Game](#) or [Deploying a Zune Game](#).

See Also [Visual Studio Differences Between Game Platforms](#)
[Game Studio Features](#)
[Developing Xbox 360 Games](#)

Visual Studio Differences Between Game Platforms

Describes the differences in the appearance and behavior of the integrated development environment (IDE) of XNA Game Studio when developing for the Xbox 360 or Zune.

- [Assembly Information Dialog Box](#)
- [XNA Game Studio Device Management Toolbar](#)
- [Add Reference Dialog Box](#)
- [Add New Item Dialog Box](#)
- [Project Properties](#)

Assembly Information Dialog Box

You can open the **Assembly Information** dialog box from the **Application** page in the Project Designer.

To open the Project Designer

1. Perform one of the following steps:
 - Under the project node in Solution Explorer, double-click the **Properties** folder.
-or-
 - From the **Project** menu, select **[project name] Properties**.
2. On the **Application page**, click **Assembly Information**.

Use this dialog box to change the assembly information for the current project. This information includes the game title, company name, copyright, and trademark, plus a brief description. When the game is installed on the target device (such as an Xbox 360 console or Zune), some of this information will be displayed in the game selection interface for that device.

The assembly information specified here only affects the values seen in the Xbox 360 Dashboard's **Game Library** if the project is a game project. Assembly information of library projects is not used when deploying or displaying information about a game.

Title

This field specifies a title for the assembly manifest, and is required to deploy a game to a game device. On most game devices, only the first 25 characters of this field are visible.

On the Xbox 360 console, the title appears in the **Game Library**.

On Zune, the title appears in the **Games** list.

Description

This field specifies an optional description for the assembly manifest. Only the first 300 characters of this field are visible.

On the Xbox 360 console, the description appears in the **Game Library**.

On Zune, the description appears after you select the game from the **Games** list.

Company

Specifies a company name for the assembly manifest.

Product

Specifies a product name for the assembly manifest.

Copyright

Specifies a copyright notice for the assembly manifest.

Trademark

Specifies a trademark for the assembly manifest.

Assembly Version

Specifies the version of the assembly.

File Version

Specifies a version number that instructs the compiler to use a specific version for the Win32 file version resource (Windows projects only).

GUID

Specifies a unique GUID that identifies the assembly. When you create a project, Visual Studio generates a GUID for the

assembly.

Neutral Language

Specifies which culture the assembly supports.

Make Assembly COM-Visible

Specifies whether types within the assembly will be accessible to COM (Windows projects only).

Platform Differences Summary

These are the differences in operation of the **Assembly Information** dialog box for specific platforms:

Platforms	Differences
Xbox 360, Zune	<ol style="list-style-type: none">1. The File Version field is not available.2. The Make Assembly COM-Visible check box is not available.

XNA Game Studio Device Management Toolbar

You can use the XNA Game Studio Device Management toolbar to maintain a list of one or more external game devices (such as Xbox 360 consoles or Zunes) available for deployment. It is available when you load a game project or game library project in supported versions of Visual Studio tools.

For more information, see:

- [Using XNA Game Studio Device Center](#)
- "Managing Several Xbox 360 Consoles with XNA Game Studio Device Center" in [Deploying an Xbox 360 Game](#)
- "Managing Several Zunes with XNA Game Studio Device Center" in [Deploying a Zune Game](#)

Platform Differences Summary

These are the differences in operation of the **Assembly Information** dialog box for specific platforms:

Platforms	Differences
Xbox 360, Zune	The XNA Game Studio Device Management toolbar is available.

Add Reference Dialog Box

Use the **Add Reference** dialog box to add component references required by your project. Because the Xbox 360 and Zune platforms do not have the same feature set as the Windows platform, the **COM** tab is hidden and the **.NET** tab contains only Xbox 360 or Zune-specific assemblies. These assemblies include:

- **Microsoft.Xna.Framework**
- **Microsoft.Xna.Framework.Game**
- **mscorlib**
- **system**
- **system.core**
- **system.xml**
- **system.xml.linq**

These assemblies are added automatically to an Xbox 360 or Zune project.

To open the **Add References** dialog box, right-click the **References** item in Solution Explorer, and then click **Add Reference**.

⚠Caution	
The Projects tab lists all projects in the solution, regardless of platform. The assemblies in the Browse and Recent tabs may also include references to assemblies that are not for the project's platform. Projects must only reference assemblies that are intended for their platform. For example, Xbox 360 projects do not support references to assemblies that target platforms other than Xbox 360, nor do Zune projects support assemblies that do not target Zune.	

Platform Differences Summary

These are the differences in operation of the **Assembly Information** dialog box for specific platforms:

Platforms	Differences
Xbox 360, Zune	<ol style="list-style-type: none">1. The COM tab is not available.2. The .NET tab contains only Xbox 360 or Zune-specific assemblies.

Add New Item Dialog Box

You can open the **Add New Item** dialog box by right-clicking the solution in Solution Explorer, or by selecting it from the **Project** menu. Use this dialog box to add new Xbox 360 or Zune-supported items. This list contains the following items.

- C# Class
- C# Interface
- C# Code File
- XML File
- XML Schema

Note

This option is available only in Microsoft Visual Studio.

- Text File
- Assembly Information File
- Resources File

Caution

The string resource type is the only built-in resource type supported by Xbox 360 or Zune projects. Adding other resource types such as bitmaps or icons to a resource file may result in compilation errors.

- Class Diagram

Note

This option is available only in Microsoft Visual Studio.

- Game Component
- Content Type Reader

The Assembly Information File template for Xbox 360 or Zune projects excludes the **AssemblyFileVersion** attribute. Neither the Xbox 360 platform nor the Zune platform support this attribute.

Project Properties

Project properties are grouped into pages in the Project Designer. You can access the Project Designer in the **Project** menu by clicking **Properties**, or by double-clicking the **Properties** item in Solution Explorer. The Project Designer property pages are located in the same middle pane used by the code editor.

- On the **Application** page, the **Target Framework** drop-down list box is disabled for Xbox 360 and Zune projects.
- On the **Build** page:
 1. The **Platform target** drop-down menu has been disabled in supported versions of Microsoft Visual Studio tools.
 2. The **Allow unsafe code** property is disabled for Zune.
- On the **Content Build** page, the **Compress content pipeline output files** check box is disabled for Zune projects.
- The **Debug** page has been modified for Xbox 360 and Zune game projects. The **Start Action** and **Enable Debuggers** options have been disabled. The **Working Directory**, **Use Remote Machine**, and **Enable the Visual Studio hosting process** controls have also been disabled.
- The **Settings** project property page allows you to add a settings file to your Xbox 360 or Zune project. However, note that the settings file is not supported for Xbox 360 or Zune projects.
- The **Security** and **Publishing** pages are hidden.

See Also [Visual Studio Unsupported Features](#)

[Game Studio Features](#)

[Developing Xbox 360 Games](#)

Project Properties

Describes the unique property settings of Project Designer in XNA Game Studio.

Topics in this section describe the special project designer properties of the XNA Game Studio development environment.

In This Section

[Application Properties Page](#)

Enables the user to change application information for the current XNA Game Studio project.

[Content Build Page, Project Designer](#)

Describes how you can use the Content Build page of the Project Designer to specify the build configuration properties for the content project.

See Also [Game Studio Features](#)

[Using XNA Game Studio](#)

Application Properties Page

Enables the user to change application information for the current XNA Game Studio project.

You open the **Application Properties** page from Project Designer. Open Project Designer by double-clicking the **Properties** item in Solution Explorer or by clicking **[project name] Properties** on the **Project** menu. Then click the **Application** tab.

For information about using most of the fields on this page, see [Application Page, Project Designer \(C#\)](#).

There is also a field specific to XNA Game Studio that lets you specify a thumbnail image for a game project, as explained below.

Setting a Game Thumbnail

The **Game Thumbnail** box enables you set the thumbnail image that appears beside your game on the game device (such as Xbox 360 or Zune) when people are selecting games to play, and on Windows when people are selecting a game to unpack. This image is distinct from the Windows icon.

There are several things to keep in mind when specifying a thumbnail for your game.

File Requirements

- The image must be saved as a .png file.
- If the largest dimension of the image you provide is not exactly 64 pixels, the image will be scaled so that its aspect ratio remains the same and its largest dimension becomes exactly 64 pixels. Such scaling can produce results you might not expect. If you provide an image whose largest dimension is already exactly 64 pixels, you can avoid scaling.
- If you do specify an image that needs to be scaled, it must be smaller than 2048 pixels by 2048 pixels in size, and the .png file that contains it must be smaller than 16 KB in size.

File Location

- The thumbnail file must be specified in the game project. Files specified in the game content subproject will not be recognized as thumbnails.
- The **Game thumbnail** drop-down menu lists all files with a .png extension that have been added to your project, regardless of each file's size or resolution.
- You can also browse for any other .png files accessible from your machine by clicking the ... button beside the **Game thumbnail** drop-down menu. When you select a file from the browse dialog box, that file is added to your project and is set as the game thumbnail.

See Also [Game Studio Features](#)

[Application Page, Project Designer \(C#\)](#)

[Developing Xbox 360 Games](#)

[Developing Zune Games](#)

Content Build Page, Project Designer

Describes how you can use the Content Build page of the Project Designer to specify the build configuration properties for the content project.

This page applies to XNA Game Studio projects only.

Content Build Compression

The following options enable you to configure settings for the build process of data in the content pipeline.

Compress content pipeline output files

Produce compressed output when processing content project data to reduce content size.

The defaults for this option vary according to the platform type and build configuration:

	Windows	Xbox 360	Zune
Debug	Off	On	Unsupported
Release	On	On	Unsupported

See Also [Content Compression](#)

[Application Properties Page](#)

[Application Page, Project Designer \(C#\)](#)

[Developing Xbox 360 Games](#)

Game Component Development

Describes how to use XNA Game Studio to develop custom game components for your game projects. You derive the new component either from the [GameComponent](#) class, or, if the component loads and draws graphics content, from the [DrawableGameComponent](#) class. For more information about existing game component support in the XNA Framework, see the Game Components section of [Application Model Overview](#).

Developing Custom Components

XNA Game Studio supports the development of custom game components for use in your game code. You can use the **Add New Item** dialog box to insert basic code for implementing a new component.

To insert a new custom game component

1. Open an XNA Framework game or library project in XNA Game Studio.
2. In Solution Explorer, right-click the game project node.
3. Click **Add**, and then click **New Item**.
4. In the **Add New Item** dialog box, in the **Categories** pane, select **XNA Game Studio 3.1**
5. In the **Templates** pane, select the **Game Component** icon.
6. Enter a name for the source file, and then click **Add**.

The new code is composed of three main methods.

Method Name	Purpose
GameComponent1	The constructor for the game component. The name matches the name you specified when you inserted the new component. Add code that initializes the component here.
Initialize	Called by the framework when the component starts. Add the component-specific starting code here.
Update	Called by the framework when the component needs to be updated. Add the component-specific update code here.

After creating the new component, add your custom code to provide the necessary functionality.

See Also [Game Studio Features Using XNA Game Studio](#)

Extending Game Studio

Describes how XNA Game Studio can be customized to support special functions to aid development of games.

This section discusses extending the special features of the XNA Game Studio development environment.

In This Section

[Accessing Custom Item Templates](#)

Describes how to access custom item templates.

[Game Studio Automation Extenders](#)

Describes DTE extenders to support automation in XNA Game Studio.

[How To: Export an XNA Game Studio Project to a Template](#)

Demonstrates how to manually modify a project template that has been exported from an existing XNA Game Studio project.

See Also [Game Studio Features](#)

Accessing Custom Item Templates

Describes how to access custom item templates. Visual Studio item templates represent items that a user can add to a project using the **Add New Item** dialog box. With a few modifications to the item template, you can make custom items represented by item templates available to XNA Game Studio projects.

Note

You can find detailed documentation of Visual Studio item templates in the Visual Studio 2008 documentation set [Creating Item Templates](#).

Item templates contain a .vstemplate file that provides an XML description of the template. For Xbox 360 and Zune projects, XNA Game Studio filters the item template list based on the contents of this file. To make an item template properly compatible with XNA Game Studio projects, add the **TemplateGroupID** element to the .vstemplate file.

Values for the **TemplateGroupID** depend on the targeted XNA Framework:

- For XNA Framework 3.0: **TemplateGroupID** can be "XnaFramework-v2," "XnaFramework-v2-Xbox360," "XnaFramework-v2-Zune," "XnaFramework-v2-Devices," and "XnaFramework-v2-Windows."
- For XNA Framework 3.1: **TemplateGroupID** can be "XnaFramework-v3.1," "XnaFramework-v3.1-Xbox360," "XnaFramework-v3.1-Zune," "XnaFramework-v3.1-Devices," and "XnaFramework-v3.1-Windows."

Note

For Windows projects, the **ShowByDefault** element will have a value of "false."

Example

The following XML example defines the **TemplateGroupID** element in the .vstemplate file of an Xbox 360 game-specific item template.

```
... <TemplateData>
  <Name>MyX360Class</Name>
  <Description>My custom Xbox 360 C# class.</Description>
  <ProjectType>CSharp</ProjectType>
  <TemplateGroupID>XnaFramework-v3.1-Xbox360</TemplateGroupID>
  <ShowByDefault>>false</ShowByDefault> ... </TemplateData> ...
```

See Also [Extending Game Studio Game Studio Features Using XNA Game Studio](#)

Game Studio Automation Extenders

Describes DTE extenders to support automation in XNA Game Studio.

This section describes the special extenders that are provided by XNA Game Studio to aid automation within the structure of an XNA game project. These extenders provide mechanisms to retrieve project objects for content projects that are subordinate to code projects, and to retrieve the code project that is parent to a content project.

Retrieving Content Projects from a Code Project

The EnvDTE.Project objects for all content projects subordinate to a code project are retrieved through the EnvDTE.Project.Extender property for that code project. The object returned implements IEnumerable, so that it can be enumerated in a For Each clause. The enumerable items implement EnvDTE.Project.

The extender objects for subordinate content projects are referenced by the string:

```
"Microsoft.Xna.GameStudio.CodeProject.NestedContentProjectsExtender"
```

The following example outputs the name of all content projects within the current code project:

Visual Basic
<pre>Sub ShowNestedContentProjects() Dim proj As Project proj = DTE.Solution.Projects.Item(1) MsgBox(proj.Name & " has extenders " & Strings.Join(proj.ExtenderNames, ", ")) For Each nestedProj As Project In proj.Extender("Microsoft.Xna.GameStudio.CodeProject.NestedContentProjectsExtender") MsgBox(proj.Name & " has nested project " & nestedProj.Name) ShowParentCodeProjects(nestedProj) Next End Sub</pre>

Retrieving Code Projects from a Content Project

The EnvDTE.Project object that is parent to any content project is retrieved through the EnvDTE.Project.Extender property for the content project. The object returned implements IEnumerable, so that it can be enumerated in a For Each clause. The enumerable items implement EnvDTE.Project.

The extender objects for parent code projects are referenced by the string:

```
"Microsoft.Xna.GameStudio.ContentProject.ParentProjectsExtender"
```

The following example outputs the name of all of a content project's parent code projects:

Visual Basic
<pre>Sub ShowParentCodeProjects(ByVal contentProj As Project) Dim parents = New System.Collections.Generic.List(Of String) For Each parentProj As Project In contentProj.Extender("Microsoft.Xna.GameStudio.ContentProject.ParentProjectsExtender") parents.Add(parentProj.Name) Next MsgBox(contentProj.Name & " has parent projects " & Strings.Join(parents.ToArray(), ", ")) End Sub</pre>

A content project may have multiple parent code projects (as in a cross-platform solutions, where content is common to all platforms).

See Also [Game Studio Features](#)

How To: Export an XNA Game Studio Project to a Template

Demonstrates how to manually modify a project template that has been exported from an existing XNA Game Studio project. It is assumed that an existing Windows or Xbox 360 game project is loaded in XNA Game Studio.

Creating an XNA Game Studio Project Template

This procedure enables you to create a template using the **Export Template** wizard, and to modify it to incorporate Game Content into the resulting template.

Note

The **Export Template** wizard of Visual Studio will create a template from an XNA Game Studio project, but it does not recognize the Content subfolder. To create a complete project template, it must be modified using the following manual process.

To create a template, you need to perform these procedures:

1. Export a game project template
2. Export a content project template
3. Merge the two templates manually into a custom template
4. Integrate the result into Visual Studio

You can perform the procedures shown here in either Visual Studio or Visual C# Express.

To create a game project template with the Export Template wizard

1. On the **File** menu, click **Export Template**.

The **Export Template** wizard opens.

2. Click **Project Template**.
3. Select the type of template that you want to create, and click **Next**.
4. Select an icon for your template.

This icon will appear in the **New Project** dialog box.

5. Enter a template name and description.
6. Uncheck the box labeled **Automatically import the template into Visual Studio**.
7. Click **Finish**.

Your project is exported into a .zip file and placed in the specified output location.

At this point in the procedure, you have created the base game project template that will be manually modified.

To create a content project template with the Export Template wizard

1. From the **File** menu, click **Close Solution**.

This closes the current solution.

2. From the **File** menu, click **Open**, and then click **Project/Solution**.
3. Type *.* in the File name field, and press **Enter**.
4. Browse to the Content subfolder of the game project that you just exported.
5. Open the Content.contentproj file to open the content project in its own solution.
6. On the **File** menu, click **Export Template**.
7. If prompted to save the solution, save it to a temporary location.

You can delete it later.

8. Using the wizard in the same manner as the previous sequence, complete the export process, and then click **Finish**.

If you did not uncheck the option to "Display an explorer window on the output files folder," an explorer window presents the folder containing the two ZIP (.zip) files created by the export process.

Next, you need to combine the two exported templates.

To extract the ZIP template

The templates created by the previous procedures are in the form of compressed ZIP files in the "My Exported Templates" folder. This procedure requires extracting the files contained in the ZIP templates, modifying the XML file that describe the project, and compressing the results to a new ZIP template.

1. From the Explorer window, extract the ZIP template for the game project to a temporary location:

Right click the .zip file, click **Extract All**, and then specify **C:\CustomTemplate** as the folder to which files will be extracted.

2. From the Explorer window, extract the ZIP template for the content project to a Content subfolder directly beneath the temporary location used in the previous step:

Right-click the .zip file, click **Extract All**, and then specify **C:\CustomTemplate\Content** as the folder to which files will be extracted.

To merge the two templates manually into a custom template

1. In Visual Studio, from the **File** menu, click **Open** to open the template (.vstemplate) file for the game project.

Specify the file **C:\CustomTemplate\MyTemplate.vstemplate**.

2. In the newly opened XML file, add the following element as a child of the <TemplateData> element:

Game Project XML File
<pre><PromptForSaveOnCreation>true</PromptForSaveOnCreation></pre>

Specifying this element will ensure that the user is prompted to name a new project when the template is used. If not specified, Visual Studio will default to creating a zero-impact project, which is not supported in XNA Game Studio.

3. Add the following XML after the <TemplateContent> element (still inside the <VSTemplate> element):

Game Project XML File
<pre><WizardExtension> <Assembly>Microsoft.Xna.GameStudio, Version=2.0.0.0, Culture=neutral, PublicKeyToken=6d5c3888ef60e27d</Assembly> <FullClassName>Microsoft.Xna.GameStudio.Wizards.NestedProjectWizard</FullClassName> </WizardExtension> <WizardData> <Folder Name="Content" TargetFolderName="Content"> <ProjectItem ReplaceParameters="true" TargetFileName="Content.contentproj">Content.contentproj</ProjectItem> <!-- Content goes here! --> </Folder> </WizardData></pre>

4. In Visual Studio, from the **File** menu, click **Open** to open the template (.vstemplate) file for the content project.

Specify the file **C:\CustomTemplate\Content\MyTemplate.vstemplate**.

5. In the newly opened XML file, locate the <Project> element.

Its child elements will consist of zero or more <ProjectItem> elements, one for each item that was in your exported Content project.

Content Project XML File
<pre><TemplateContent> <Project TargetFileName="Content.contentproj" File="Content.contentproj" ReplaceParameters="true" /> <ProjectItem ReplaceParameters="false" TargetFileName="Miramonte.spritefont">Miramonte.spritefont</ProjectItem> </Project> </TemplateContent></pre>

- Select the <ProjectItem> elements from the content project file, and click **Copy**.
- Paste the <ProjectItem> elements into the game project file—replacing the comment <!-- Content goes here! -->— and then click **Save**.

Game Project XML File

```
<WizardExtension>
  <Assembly>Microsoft.Xna.GameStudio, Version=2.0.0.0, Culture=neutral, PublicKeyToken=6d5c3888ef60e27d</Assembly>
  <FullClassName>Microsoft.Xna.GameStudio.Wizards.NestedProjectWizard</FullClassName>
</WizardExtension>
<WizardData>
  <Folder Name="Content" TargetFolderName="Content">
    <ProjectItem ReplaceParameters="true" TargetFileName="Content.contentproj">Content.contentproj
  </ProjectItem>
    <ProjectItem ReplaceParameters="false" TargetFileName="Miramonte.spritefont">Miramonte.spritefont
  </ProjectItem>
  </Folder>
</WizardData>
```

- In Explorer, browse to the folder where you extracted the game template (for example, C:\CustomTemplate).
- Press **CTRL+A** to select all files and folders in the directory.

Note

Do not select the CustomTemplate directory itself.

- Right-click the selection, and then click **Send To** and **Compressed (zipped) folder** to create a .zip file. It should be created in the same directory. The resulting file is your new, custom project template.

To integrate the custom template into Visual Studio

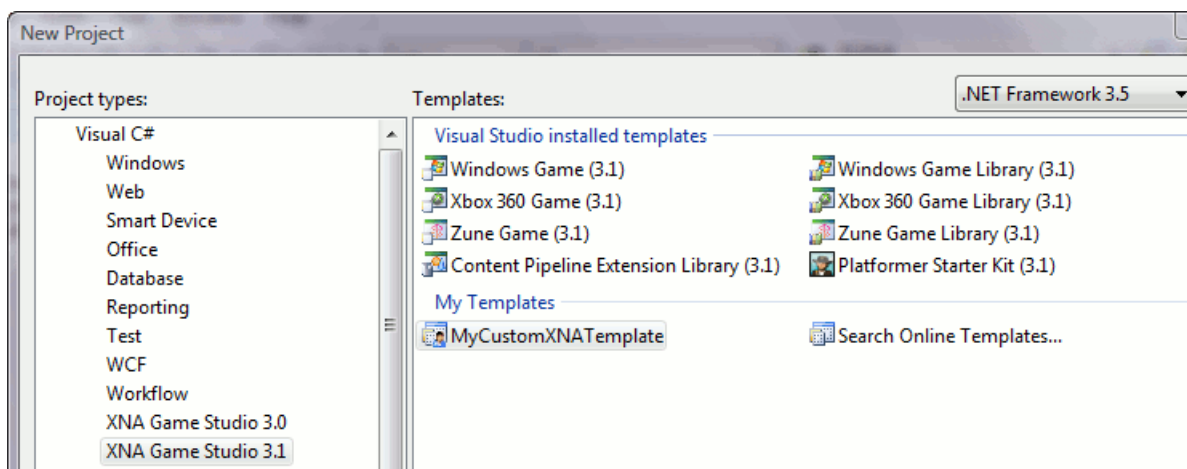
First, you need to find the path for your user templates folder.

- From the **Tools** menu, click **Options**.
- In the resulting dialog box, click the **Projects and Solutions** node.

The path is identified in the **Visual Studio user project templates location** setting. The default setting is a subfolder named ProjectTemplates.

- Copy the .zip file into a subdirectory called Visual C#\XNA Game Studio 3.1\.

When you complete this lengthy procedure correctly, your custom template will appear below the standard XNA Game Studio projects in the **New Project** dialog box. You can use it now.



See Also [Extending Game Studio Game Studio Features](#)
[Using XNA Game Studio](#)

Sharing and Distributing Your Game

When you've created something incredible and want to show it off to others, XNA Game Studio makes it easy.

The XNA Creators Club offers the best way to share your game with a wide community of gamers, to get feedback and build enthusiasm for your game.

There are a variety of ways to share and distribute your game.

- For all platforms, you can use the XNA Game Studio Package utility (xnapack.exe) to create a special package that other XNA Game Studio users can unpack and play. Using this utility allows you to share your game without exposing your source code or assets to other XNA Game Studio users.

You can share games packaged in this format with [XNA Creators Club](#) members through the [XNA Creators Club Web site](#) for peer review, and possible listing on Xbox LIVE Marketplace.

- For all platforms, you can choose to share your source code and assets with other XNA Game Studio users. This is useful if you would like someone to review or modify your code or assets.
- For Windows games only, you can make your game available to any Windows user by distributing your completed game executables, along with a set of prerequisite software (called "redistributables"). All Windows users can play your game, even if they do not have XNA Game Studio installed on their computer.

The following are the available sharing options for Windows, Xbox 360 and Zune.

	Sharing Your Game Package	Sharing Your Source Code and Assets	Distributing Your Finished Windows Game
Windows	...with XNA Game Studio users	...with XNA Game Studio users	...with any Windows user
Xbox 360	...with XNA Creators Club members ...through Xbox LIVE Marketplace (after a approval)	...with XNA Creators Club members	
Zune	...with XNA Game Studio users	...with XNA Game Studio users	

See the sections below for more information on each sharing and distribution type, including how to prepare your game to be shared or distributed, and how to play a game someone has shared with you.

In This Section

[Sharing Your Game Package](#)

Describes how to distribute your game to other XNA Game Studio users in a package that protects your source code and assets.

[Sharing Your Source Code and Assets](#)

Presents information about sharing your game's source code and assets with other XNA Game Studio users.

[Distributing Your Finished Windows Game](#)

Presents information about how to share your game with any Windows user, even if the user does not have XNA Game Studio installed.

Sharing Your Game Package

Describes how to distribute your game to other XNA Game Studio users in a package that protects your source code and assets.

Quick Information

Platform	Source Code Visible	Format	Windows User Requirements	Xbox 360 User Requirements	Zune User Requirements
Windows, Xbox 360 and Zune	No	.ccgame XNA Creators Club Game Package	<ul style="list-style-type: none"> Must have XNA Game Studio installed. 	<ul style="list-style-type: none"> Must have XNA Game Studio installed. Must have an active membership in the XNA Creators Club. 	<ul style="list-style-type: none"> Must have XNA Game Studio installed.

If you would like to share your XNA Game Studio game with other XNA Game Studio users, but do not want to share your source code and assets, you can use the XNA Game Studio package utility (xnpack.exe). You can use the package utility from within supported versions of Microsoft Visual Studio tools, or from the command line.

You can share games packaged in this format with [XNA Creators Club](#) members through the [XNA Creators Club Web site](#) for peer review and possible listing on the Xbox LIVE Marketplace.

Note
Currently, only Xbox 360 games can be shared through XNA Creators Club.

There are two fundamental ways that a game package can be used:

- [Sharing a Game with Another User](#)
- [Sharing a Game from Another User](#)

Sharing with Another User

The XNA Game Studio package utility compresses your game's built executable and assets into a special compressed format that other XNA Game Studio users can open and play. You may package only XNA Game Studio game projects. The package utility does not support XNA Game Studio library projects.

To package your game using the XNA Game Studio package utility from the Visual Studio development environment

1. In XNA Game Studio, open the project you want to package.
2. On the **Build** menu, click **Package [projname] as XNA Creators Club Game**, where [projname] is the name of the current project.

The output window indicates the status of the packing operation. If the packing operation is successful, the package file will be created with a .ccgame extension, and placed in the appropriate output folder of the project. For example, the package file for the release version of your Windows game would be located in the bin\x86\Release folder. This example assumes that the default project settings have not been modified.

You may share this .ccgame file with other XNA Game Studio users. It is a compressed version of your project that contains no source code or source assets. Other users can play it, but they will not be able to view or modify your source files.

To package your game using the XNA Game Studio package utility from the command line

1. To open an XNA Game Studio command line, open the **Start** menu, click **XNA Game Studio 3.1**, click **Tools**, and then click **XNA Game Studio Command Prompt**.
2. From the command line, use the **cd** command to change the active folder to the appropriate output folder of the project you want to package.
3. Enter a packing command using the following syntax:

```
xnpack assembly [platform] [/nologo] [/output:file] [/quiet] [/thumbnailfile]
```

The command-line parameters are:

Parameter	Usage
<i>assembly</i>	The startup assembly of your project. This is the .exe built by your XNA Game Studio project.
[<i>platform</i>]	Deprecated and optional. Specifies the platform for which the project is built. This value can be "Windows," "Xbox360," or "Zune," and must match your game's project type.
[/nologo]	Suppresses logo/copyright messages.
[/output:file]	Specifies the path and file name of the output file. By default, your game is created in the same folder as the startup assembly with the same name. (Short form: /o:)
[/quiet]	Suppresses all non-error messages.
[/thumbnail:file]	Specifies the path to an optional thumbnail that is displayed with your game. (Short form: /th:) For a Windows game, this thumbnail appears in the graphical unpacking utility. For an Xbox 360 game, the thumbnail also appears in the Game Library . For a Zune game, the thumbnail appears in the Games list.

The following example packs a Windows game called `MyWinGame` and suppresses the logo/copyright messages.

```
xnapack MyWinGame.exe /nologo
```

The Command-line window indicates the current status of the packing operation. If the packing operation is successful, you created the package file, and it is in the same folder as the startup assembly, with a `.ccgame` extension.

After you packed a game, you may share the `.ccgame` file with other XNA Game Studio users. It is a compressed version of your project that contains no source code or source assets. Other users can play it, but they will not be able to view or modify your source files.

Sharing from Another User

When you receive a `.ccgame` file from another XNA Game Studio user, you can use either the graphical unpacking utility or the command-line utility to unpack it.

Consider the following before unpacking.

- You must have XNA Game Studio installed.
- Unpacking a `.ccgame` file automatically deploys the assets and game assembly to the target platform.
 - For a Windows project, you must have the same version of the XNA Framework installed with which the `.ccgame` file was built.
 - For an Xbox 360 project, you must have at least one Xbox 360 console specified already in **XNA Game Studio Device Center**, you must have a valid XNA Creators Club membership, and you must have XNA Game Studio Connect running on the Xbox 360 console to which you are deploying the assets and game assembly.
 - For a Zune project, you must have at least one Zune digital media device specified in **XNA Game Studio Device Center**.

To unpack a packaged game from Windows Explorer

1. From Windows Explorer, double-click the `.ccgame` file you want to unpack.

A dialog box appears.

2. Click **Unpack** to begin the unpacking process.
3. When the unpacking process is complete, the next step depends on the platform.
 - For a Windows project, when the unpacking is complete, a folder containing the game executable opens in Windows Explorer. Run the game by double-clicking the executable.
 - For an Xbox 360 project, when the unpacking is complete, you can run the game from the **Game Library**.
 - For a Zune project, when the unpacking is complete, you can run the game from the **Games** list.

 **Caution**


If the intended destination of an unpacked game already exists, the XNA Game Studio package utility deletes the existing destination and its contents. It then creates a new container or folder containing the contents of the new game package. This applies to games targeting Xbox 360, Zune, and Windows platforms.

To unpack a packaged game from the command line

1. To open an XNA Game Studio command prompt, open the **Start** menu, click **XNA Game Studio 3.1**, click **Tools**, and then click **XNA Game Studio Command Prompt**.
2. From the command line, use the **cd** command to change the active folder to the folder containing the appropriate package.
3. Enter an unpacking command using the following syntax.

```
xnapack unpack package [/listplatforms] [/listtargets] [/nologo] [/platform:name] [/quiet] [/run] [/showui]
[/target:name]
```

The command-line parameters are:

Parameter	Usage
<i>package</i>	An XNA Framework game package, previously created with the XNA Game Studio package utility. If the game is intended for the Xbox 360 console, see Connecting to Your Xbox 360 Console with XNA Game Studio 3.1 for information about adding registered consoles to your computer. If the game is intended for the Zune digital media device, see Connecting to your Zune Device with XNA Game Studio for information about adding registered devices to your computer.
/listplatforms]	Lists the platforms that support the XNA Creators Club game package.
/listtargets]	Lists the platform instances for a given platform. Must be used with /platform option.
/nologo]	Suppresses logo/copyright messages.
/platform:<i>name</i>]	Specifies the platform to which the game package is being deployed. Required if the package supports more than one platform and the target cannot be resolved.
/quiet]	Suppresses all non-error messages.
/run]	Runs the game once installation is complete. Applies only to Xbox 360 and Zune games.
/showui]	Displays the user interface for confirming and copying files.
/target:<i>name</i>]	Specifies the named platform instance to which the game package is being deployed. If no platform instance is specified, the default target is used. For a list of available targets, use the /listtargets option.  Note This option replaces the obsolete option /console .

The following example unpacks an XNA Creators Club game package containing a Windows game called `MyWinGame`, and suppresses all non-error messages.

```
xnapack unpack MyWinGame-Windows.cgame /quiet
```

4. When the unpacking process is complete, the next step depends on the platform.
 - For a Windows project, when the unpacking is complete, a folder containing the game executable opens in Windows Explorer. Run the game by double-clicking the executable.
 - For an Xbox 360 project, when the unpacking is complete, you can run the game from the **Game Library** on your Xbox 360 console.
 - For a Zune project, when the unpacking is complete, you can run the game from the **Games** list on your Zune device.

Caution

If the intended destination of an unpacked game already exists, the XNA Game Studio package utility deletes the existing destination and its contents. It then creates a new container or folder containing the contents of the new game package. This applies to games targeting Xbox 360, Zune, and Windows platforms.

See Also [Sharing Your Source Code and Assets](#)
[Distributing Your Finished Windows Game](#)

Sharing Your Source Code and Assets

Presents information about sharing your game's source code and assets with other XNA Game Studio users.

Quick Information

Platform	Source Code Visible	Format	Windows User Requirements	Xbox 360 User Requirements	Zune User Requirements
Windows, Xbox 360 and Zune	Yes	.zip or other compressed format	<ul style="list-style-type: none"> Must have XNA Game Studio installed. 	<ul style="list-style-type: none"> Must have XNA Game Studio installed. Must have an active membership in the XNA Creators Club. 	<ul style="list-style-type: none"> Must have XNA Game Studio installed.

Sharing your entire project, including source code and content, is useful when you want other users to review your code or content, or make modifications. However, in some cases you may not want others to see your source code or assets. If you want feedback on gameplay or existing game art, and don't explicitly need your code to be reviewed, consider the method described in [Sharing Your Game Package](#).

Sharing to Another User

To share your project, including source code, with an XNA Game Studio user

- In XNA Game Studio, open the project you want to share.
- On the **Build** menu, click **Clean Solution**.
This removes any built and intermediate files, leaving only the source code and assets. If you do not have this menu item, see "Cleaning the Solution Before You Build" in [Deploying an Xbox 360 Game](#).
- Open Windows Explorer, and browse to the folder that contains the project you want to share.
- Right-click on the project folder in Windows Explorer, click **Send To**, and then click **Compressed (zipped) Folder**.

The compressed file is ready to share.

Sharing from Another User

To open a project with source code as an XNA Game Studio user

- Upon receiving a compressed file with source code and assets, extract the contents into a folder.

Tip

If it is a .zip file, you can right-click the file in Windows Explorer, click **Open With**, and then click **Compressed (zipped) Folders** to begin the extraction process.

- After the files are extracted, start XNA Game Studio.
- In XNA Game Studio, click the **File** menu, and then click **Open Project**.
- Use the dialog box to browse to the project folder you specified in Step 1, and select the .csproj file.
- Click **Open**.

The project is loaded into XNA Game Studio and ready for viewing or modification.

See Also [Sharing Your Game Package](#)
[Distributing Your Finished Windows Game](#)

Distributing Your Finished Windows Game

Presents information about how to share your game with any Windows user, even if the user does not have XNA Game Studio installed.

Quick Information

Platform	Source Code Visible	Format	Windows User Requirements
Windows	No	Setup.exe and manifest, or .zip or other compressed format	<ul style="list-style-type: none"> • Must be running at least Windows XP Service Pack 2. • Must have a video card that supports at least Shader Model 1.1.

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GameServicesNotAvailableException](#).

Also, the XNA Framework Redistributable file does not contain the [Content Pipeline Build Runtime](#). Building content at run time is supported only when XNA Game Studio has been installed on the Windows-based development computer.

If you would like to share your XNA Game Studio game with other Windows users, but do not want to share your source code and assets, you can create an installable package that contains your game binaries and the necessary redistributable files.

The ClickOnce publishing feature of Visual Studio provides a convenient way to produce a portable installation package for distribution on a CD, Web site, or other media.

Sharing With Another Windows User

You may package XNA Game Studio game projects for use without XNA Game Studio installed only for the Windows platform.

You can use one of the following methods to create a portable install package:

Using ClickOnce Deployment

You can use the [ClickOnce](#) feature of Visual Studio to build and create an installation package for your XNA Game Studio game. This method automatically ensures that all required supporting files are included in your installation package.

Using Another Installation Package Tool

You can use an installation package tool of your choice. With this method, you must take more care to include all of the required supporting files for an XNA Game Studio game.

Using ClickOnce Deployment

ClickOnce is a deployment technology that allows you to create self-updating Windows-based applications that can be installed and run with minimal user interaction.

To build a ClickOnce installation package

1. In **Solution Explorer**, select the game project for the Windows platform to which you want to deploy.
2. From the **Build** menu, click **Publish**.
The Publish Wizard appears.
3. In the **Where do you want to publish the application?** page, enter the file path or FTP location where the application will be deployed.
The default location is a folder named "publish" beneath the project's folder.
4. Click **Next** to continue.
5. In the **How will users install the application?** page, select the **From a CD-ROM or DVD-ROM** option if that is your choice (the default), and then click **Next** to continue.
6. In the **Where will the application check for updates?** page, choose an update option:
 - If the application checks for updates, click **The application will check for updates from the following**

Location, and enter the location where updates will be posted. This can be a file location, Web site, or FTP Server.

- If the application will not check for updates, click **The application will not check for updates**.
- Click **Next** to continue.

7. Click **Finish** to deploy the application.

Visual Studio builds your XNA Game Studio game project and, if successful, deploys your application to the location specified in step 3. A Windows Explorer window then opens to show the Setup.exe file produced and the manifest.

The ClickOnce feature for XNA Game Studio automatically includes all of the required software packages as part of the installation package.

An alternate method of initiating ClickOnce is through the **Publish** properties of the Project Designer. These can be accessed by double-clicking the **Properties** item in Solution Explorer or by clicking **[project name] Properties** on the **Project** menu. Then click the **Publish** tab. This tab provides the facility to set several more options, including the version number of the deployed application.

If you wish to distribute your game on a CD, use a CD-Rewriter or DVD-Rewriter to copy the files from the specified location to the CD-ROM or DVD-ROM media.

Special Considerations for Game Data Files

Your game may require the use of a data file to read and store information. For example, the starter kit provided with XNA Game Studio uses an XML file to read and store game settings. Other games may use a text file or other format.

The default behavior of the ClickOnce publishing wizard assigns any XML, text or other data files in a project to "Data File" as its Publish Status. However, this status is for files containing data used by the installer package, not for files accessible to your game when it executes. Files assigned to this type will be placed in the path denoted in the .NET Framework's ApplicationDeployment.DataDirectory property.

This is likely to be inconvenient, as the file will not be accessible to your game through conventional storage APIs. The recommended actions to correct this are as follows:

1. Ensure that the **Build Action** property of the data file is set to "Content." This specifies that the data file is to be included in the install package.
2. Reassign the Publish Status to "Include."

The **Build Type** setting may be changed through the file's property window.

The Publish Status is changed through the **Application Files** dialog box of the **Publish** properties section of Project Designer.

To change the Publish Status

- Double-click the **Properties** item of your project in Solution Explorer.
- Navigate to the **Publish** tab in Project Designer.
- Click the **Application Files** button.
- Select the drop-down box in the **Publish Status** column of the dialog box to change the setting.
- Click **OK**.

Using Another Installation Package Tool

To use an installation package tool other than ClickOnce, you will need to build your game binaries and include the required software packages.

To compile your game binaries for distribution with another installation package tool

1. In XNA Game Studio, open the project you want to package.
2. From the **Build** menu, click **Build Solution**.

This creates the necessary files needed to run your game on other computers.

3. Using an installation package tool of your choice, package all files located in the appropriate output directory of your project.

Required Software Packages

In addition to the requirements listed in the previous table, there are additional software prerequisites that must be installed on a player's machine if the computer does not have XNA Game Studio installed.

The following software packages must be installed on the player's computer in order for your game to run.

- The [.NET Framework 3.5 Redistributable](#).
- The [XNA Framework Redistributable 3.1](#).

Tip

You can verify that your game was built with the XNA Framework 3.1 by loading your project in XNA Game Studio, and clicking the project node in Solution Explorer. The **XNA Framework Version** property is visible in the property pane.

To determine whether the player already has the redistributable installed, use the Windows Installer function [MsiQueryProductState](#) with the product code for the version of the redistributable that you want to verify. The following table lists the product code for the XNA Framework Redistributable 3.1.

Redistributable Version Corresponding To...	Product Code
XNA Game Studio 3.1 (English)	19BFDA5D-1FE2-4F25-97F9-1A79DD04EE20
XNA Game Studio 3.1 (Japanese)	F037A396-7FA3-4FB4-ACB8-3C6FE57B02BD

An alternative to checking the product code for the version of the XNA Framework Redistributable for XNA Game Studio 3.1 is to check the registry for the following value:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\XNA\Framework\v3.1]
Installed=1
```

- The following files from the [DirectX 9.0c Redistributable](#).
 - OCT2006_d3dx9_31_x86.cab
 - APR2007_d3dx9_33_x86.cab
 - APR2007_xinput_x86.cab
 - Mar2009_xact_x86.cab
 - Mar2009_XAudio_x86.cab
 - Mar2009_X3DAudio_x86.cab
 - DSETUP.dll
 - dsetup32.dll
 - DXSETUP.exe
 - dxupdate.cab

These files are installed automatically as part of the XNA Framework Redistributable install package for XNA Game Studio 3.1. You don't need to follow a separate install step when you create games with this version of XNA Game Studio.

See Also [Sharing Your Source Code and Assets](#)
[Sharing Your Game Package](#)

Programming Guide

Describes how to use the XNA Framework to develop games in XNA Game Studio. The XNA Framework is a set of managed libraries for Windows, the Xbox 360, and Zune. These libraries enable you to be more productive by using a set of unified class libraries to develop C# games.

How To Get Started

If you are new to game programming, or just want to review the basic steps toward getting a simple game up and running in XNA Game Studio, see the how-to articles in [Game Programming Basics](#).

Extended Tutorials

To see how to integrate XNA Framework features and follow best practices for creating a complete game, see the articles in [Extended Tutorials](#).

How-to articles are available in the following categories.

In This Section

[Game Programming Basics](#)

To get started writing games in XNA Game Studio, you may wish to review these key how-to topics.

[Content Pipeline](#)

Provides an overview of the Content Pipeline. The XNA Game Studio Content Pipeline builds art assets that you have included in your project into a form your game can load at run time on either Windows or the Xbox 360 game machines by calling `ContentManager.Load`.

[Application Model](#)

Provides functionality to accomplish common game development tasks.

[Graphics](#)

Describes how the XNA Framework Graphics libraries provide low-level resource loading and rendering capabilities.

[Math](#)

Provides classes and methods for manipulating vectors and matrices.

[Input](#)

Provides classes and methods for retrieving user input for keyboard, mouse, and Xbox 360 controller devices.

[Audio](#)

Provides classes and methods for playing audio files.

[Media](#)

Describes how the XNA Framework `Microsoft.Xna.Framework.Media` namespace provides classes and methods for retrieving system media, including pictures and songs.

[Storage](#)

Provides classes that allow reading and writing of files.

[Gamer Services](#)

Contains introductory articles describing how to use gamer services: working with player profiles and preferences, the Xbox Guide user interface, Guide-based messaging, and other features provided by Xbox LIVE.

[Networking](#)

Contains introductory articles describing how to create and join multiplayer game sessions, manage game state across clients, and interact with the friends list.

[Hardware and Platforms](#)

Provides information about programming for specific hardware types and platforms using the XNA Framework.

[Extended Tutorials](#)

Describes how to integrate XNA Framework features and follow best practices for creating games.

Game Programming Basics

To get started writing games in XNA Game Studio, you may wish to review these key how-to topics.

Fundamental How-To Topics

In This Section

[Adding Game Assets to Your Game](#)

Demonstrates how to add a texture asset to your game. The same procedure can also be applied to model and sound assets.

[How To: Load Content](#)

Demonstrates how you can load content and ensure that the content will be reloaded at the appropriate times.

[How To: Draw a Sprite](#)

Demonstrates how to draw a sprite by using the [SpriteBatch](#) class.

[How To: Draw Text](#)

Demonstrates how to import a [SpriteFont](#) into a project and draw text using [DrawString](#).

[How To: Make a First-Person Camera](#)

Demonstrates how to create a first-person camera.

[How To: Render a Model](#)

Demonstrates how to load and render a model using the XNA Framework Content Pipeline.

[How To: Use BasicEffect](#)

Demonstrates how to create and initialize an instance of [BasicEffect](#), initialize a vertex buffer that can be rendered by [BasicEffect](#), apply the effect, and render the geometry.

[How To: Detect Whether a Controller Button Is Pressed](#)

Demonstrates how to detect whether a user has pressed a digital button on a connected Xbox 360 Controller.

[How To: Play a Sound](#)

Demonstrates how to play a sound.

[How To: Write Games for Less Capable Hardware](#)

Demonstrates how to write games for a variety of computer configurations. Specifically, this topic addresses ways to write a game so it will work on high-end hardware, yet continue to give adequate performance when the game is run on a computer that is less capable.

[How To: Open a File](#)

Demonstrates how to use the [StorageContainer](#) class to open a save game file in the title storage area on a device specified by the gamer.

Related Topics

For step-by-step guidance through creating a simple game, see the following sections.

[Your First Game: Microsoft XNA Game Studio in 2D](#)

[Going Beyond: XNA Game Studio in 3D](#)

Content Pipeline

Provides an overview of the Content Pipeline. The XNA Game Studio Content Pipeline builds art assets that you have included in your project into a form your game can load at run time on either Windows or the Xbox 360 game machines by calling [ContentManager.Load](#).

This build process for art assets is controlled by Content Pipeline importers and content processors. When you press F5 to build a game created with XNA Game Studio, the appropriate Content Pipeline importer and processor for each asset is invoked, and that asset is automatically built into your game.

The flexibility of this process enables you create and update art assets using a wide variety of digital content creation (DCC) tools. XNA Game Studio supplies importers for several popular export formats supported by DCC tools, and also lets you develop custom importers for other formats.

In This Section

[Overview of the Content Pipeline](#)

Describes how the XNA Game Studio Content Pipeline lets you build art assets into your game automatically from the file formats in which they are maintained.

[Model Processing with the XNA Framework Content Pipeline](#)

Describes the model conversion process implemented by XNA Framework Content Pipeline.

[Content Pipeline Architecture](#)

Describes the architecture of the XNA Game Studio Content Pipeline build process. The process is designed to be extensible, so that it can easily support new input file formats and new types of conversion.

[Content Compression](#)

XNA Game Studio offers data compression - an easy way to decrease the size of certain built game assets (e.g. textures, shaders, and meshes). This reduces the size required for media storage and downloads, and it reduces the deploy time required when debugging a game.

[Standard Importers and Processors](#)

Describes the standard Content Pipeline importers and content processors of XNA Game Studio that support various common art-asset file formats.

[Parameterized Processors](#)

Describes how parameterized processors work in XNA Game Studio. Many of the standard Content Pipeline content processors shipped with XNA Game Studio support parameter usage.

[Developing with Parameterized Processors](#)

Describes how developing with parameterized processors, both standard and custom, requires additional thought and care.

[Using a Custom Importer or Content Processor](#)

Describes how to use a custom processor or importer in an existing game solution.

[Extending an XNA Framework Standard Processor](#)

Describes how XNA Game Studio lets you modify or extend the behavior of any of the standard Content Pipeline processors that ship with the product.

[Sprite Font XML Schema Reference](#)

Describes the valid tags and values for Sprite-Font (.spritefont) XML files used by the Content Pipeline to create [SpriteFont](#) textures.

[How To: Write a Custom Importer and Processor](#)

Describes how a content importer adds support for a new art asset file format.

[How To: Extend the Font Description Processor to Support Additional Characters](#)

Describes the process of developing a custom content processor needed to add additional characters to a [FontDescription](#) object based on the text that is required by the game.

See Also

Concepts

[Programming Guide](#)

[Content Pipeline Content Catalog at XNA Creators Club Online](#)

Overview of the Content Pipeline

Describes how the XNA Game Studio Content Pipeline lets you build art assets into your game automatically from the file formats in which they are maintained.

Most games use art in the form of models, meshes, sprites, textures, effects, terrains, animations, and so on. Such art assets can be created in many different ways and stored in many different file formats. They tend to change frequently in the course of game development.

The Content Pipeline is designed to help you include such art assets in your game easily and automatically. For example, an artist working on a car model can add the resulting file to the XNA Game Studio game project, assign the model a name, and choose an importer and content processor for it. Then a developer who wants to make the car drive can load it, by name, using a call to [ContentManager.Load](#). This simple flow enables the artist to focus on creating assets and the developer to focus on using them, without either having to spend time worrying about content transformation.

Purpose of the Content Pipeline

The XNA Framework Content Pipeline is designed to:

- Enable game artists to use the digital content creation (DCC) tools of their choice.
- Provide a mechanism to decouple digital content's dependency on a particular game engine.
- Provide a simple, expandable content build system that meets the needs of both artists and developers.

Basics of the Content Pipeline

Recognizing that DCC tools save content in many different file formats, the XNA Game Studio Content Pipeline enables you build art assets into your game automatically from the file formats in which they are maintained. Here's a high-level view of how it works.

- XNA Game Studio supplies standard importers and processors for a number of popular DCC file formats (see [Standard Importers and Processors](#)).
- Third parties also create custom importers and processors for XNA Game Studio to support additional formats.
- If you have enough information about a DCC file format, you can write your own custom importer and processor for it using classes provided by the Content Pipeline class library (see [How To: Write a Custom Importer and Processor](#)).
- When you include an art asset file in your XNA Game Studio game project, you use its Properties sheet to specify the appropriate importer and processor. Thereafter, when you press F5 to build your game, the proper importer and processor for each asset is invoked automatically. The asset is built into your game in a form that can be loaded at run time on Windows or the Xbox 360 by using [ContentManager.Load](#).

Importers vs. Content Processors

- An *importer* takes art assets saved in a particular DCC file format and converts them into objects in the XNA Game Studio Content DOM (document object model) that standard content processors can consume, or into some other custom form that a particular custom processor can consume.
- A *processor* takes one specific type of imported art asset, such as a set of meshes, and compiles it into a managed code object that can be loaded and used by XNA Game Studio games on Windows and the Xbox 360.

Automatic Serialization of .XNB Files

Custom content types are processed by the Content Pipeline with a custom importer and run-time processor, which are supplied by you or by an external developer. Part of the process is handled by the Content Pipeline's XNB Serializer. It is responsible for writing to and reading from the intermediate format (.XNB) used by XNA GS. Before XNA GS 3.1, a custom writer (implemented by a user-defined class based on [ContentTypeWriter](#)) was needed to write the custom content data to the .XNB file. In addition, a custom reader (implemented by a user-defined class based on [ContentTypeReader](#)) was needed to read the custom content data from the .XNB file and initialize the proper type with that data.

Starting with XNA Game Studio 3.1, the serialization of custom data to the .XNB format is done automatically for simple types that do not have an existing content type writer. This means that it is no longer necessary to implement a separate writer and reader class for each custom data type.

 **Note**

If your run-time class matches the layout of the design-time class, the Content Pipeline XNB Serializer automatically recognizes the relationship of the two types, and the run-time object is properly loaded with the serialized data. However, if the two differ in a significant way, the [ContentSerializerRuntimeType](#) attribute must be applied to the design time type. This specifies the correct class to initialize with the serialized data.

For instance, the previous version of the SpriteSheet sample implemented a custom writer (called `SpriteSheetWriter`) and a custom reader type (`SpriteSheetReader`). These classes serialized the output data from the processor into XNB format (`SpriteSheetWriter`), and deserialized that same data at run time (`SpriteSheetReader`) into a `SpriteSheet` object.

As mentioned previously, if the output type from the Content Pipeline differs from the run-time type, you will need to use the [ContentSerializerRuntimeType](#) attribute. This is demonstrated in the updated SpriteSheet sample, available on the Creator's Club [Web site](#). The `SpriteSheetContent` class declaration has the following line of code:

```
[ContentSerializerRuntimeType("SpriteSheetRuntime.SpriteSheet, SpriteSheetRuntime")]
```

This tells the serializer into which type the content should be loaded when `ContentManager.Load` is called. This is required because the texture field differs in type between the two classes: `Texture2D` versus `Texture2DContent`.

Note

If you move the run-time `SpriteSheet` class into a different namespace or assembly, you must also update the type's location, which is specified in the [ContentSerializerRuntimeType](#) attribute.

In addition to the [ContentSerializerRuntimeType](#) attribute, you can also version-type your custom content types by applying the [ContentSerializerTypeVersion](#) attribute.

See Also

Concepts

[Content Pipeline](#)

[Content Pipeline Architecture](#)

[Content Pipeline Content Catalog at XNA Creators Club Online](#)

Model Processing with the XNA Framework Content Pipeline

Describes the model conversion process implemented by XNA Framework Content Pipeline.

The conversion of a game asset in your project is a complex and detailed process comprised of many steps. The detailed description of the general process can be found [here](#). However, it can also be helpful to follow the process from the perspective of a single game asset type—specifically, a model. This discussion focuses on one section (a horizontal cross-section, starting with the top asset group) of the Content DOM graphic, shown [here](#).

Output from the Content Importer

The content pipeline content DOM represents an exported scene (from a modeling package) as a hierarchy of [NodeContent](#) objects. This means that a content importer designed to import a 3D scene or model must convert a 3D file into a [NodeContent](#) hierarchy containing a parent, a transform, and a collection of children. Some of these nodes will be of type [MeshContent](#), derived from [NodeContent](#). They represent the 3D objects in the scene with a mesh of triangles.

A [MeshContent](#) object is composed of a collection of [GeometryContent](#) objects. All triangles within a [GeometryContent](#) object have the same material applied to them. Their vertices contain the same type of per-vertex data, such as normals and texture coordinates. Since each [GeometryContent](#) collection is contained by a single [MeshContent](#) object, they share the same world transform.

Each [GeometryContent](#) object contains a [VertexContent](#) object (storing the values of all per-vertex data channels, and also indices into the shared [Positions](#) collection in the parent [MeshContent](#)) and an [IndexCollection Class](#), containing indices into the [VertexContent](#) object.

The content pipeline content DOM represents a model material by the [MaterialContent](#) class. [MaterialContent](#) has two derived classes often used by the content pipeline: [BasicMaterialContent](#) and [EffectContent](#). Think of [BasicMaterialContent](#) as the content pipeline equivalent of [BasicEffect](#). In the same vein, [EffectContent](#) is the content pipeline equivalent of [Effect](#).

Output from the Model Processor

The scene hierarchy (previously created by the content importer) is the input of the model processor (implemented by [ModelProcessor](#)) and is converted into a format usable by your game at run time. Optimization, such as reordering mesh triangles to maximize cache coherency, is also performed at this time. The conversion of content importer output to a run-time type or types, is the main goal of the model processor.

The types output by [ModelProcessor](#) are close to the final XNA Framework run-time types. For example, the [ModelProcessor](#) output type [ModelContent](#) corresponds to the XNA Framework run-time type [Model Class](#). This class is similar to the run-time type but stores the model data as simple managed objects, rather than GPU data types. This approach avoids the instantiation of actual GPU objects during the XNA Game Studio Content Pipeline build process. This is essential when building graphics for the Xbox 360 platform because instantiation of these types on the Windows GPU during the build process would not be feasible.

Conversion of the scene hierarchy is broken down as follows:

The entire scene hierarchy, represented by a root [NodeContent](#) and its children, is converted to a [ModelContent](#) object.

Note

Children of type [NodeContent](#) are turned into [ModelBoneContent Class](#) objects, and children of type [MeshContent](#) are turned into [ModelMeshContent](#) objects.

All [GeometryContent](#) objects, containing the actual triangles, are converted to [ModelMeshPart](#) objects. As mentioned previously, a [GeometryContent](#) object contains a [VertexContent](#), [IndexCollection](#), and [MaterialContent](#) object. Although the [ModelProcessor](#) modifies the data in the [IndexCollections](#), it does not change the type. The [VertexContent](#) object, received as input, has two corresponding output types: a [VertexBufferContent Class](#), containing the optimized triangle data, and an array of [VertexElement Structure](#) objects, specifying the data contained in the [VertexBufferContent Class](#).

The last step hands off any [MaterialContent](#) objects in the scene by chaining to the [MaterialProcessor](#).

See Also [Content Pipeline](#)
[Content Pipeline Architecture](#)

Content Pipeline Architecture

Describes the architecture of the XNA Game Studio Content Pipeline build process. The process is designed to be extensible, so that it can easily support new input file formats and new types of conversion.

While most users of the Content Pipeline can ignore its inner workings, if you are a game developer who wants to create a new importer and processor to support a new file format or game-engine capability, it is useful to understand the stages that the Content Pipeline passes through as an asset is transformed from a digital-content creation (DCC) output file to part of the game binary.

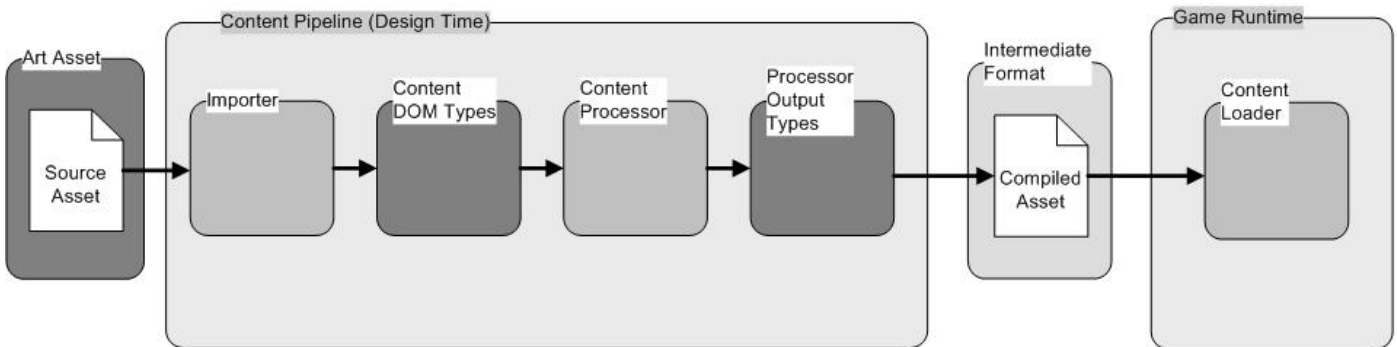
Build-Management Functionality

Once an art asset (such as a car model) is added to an XNA Game Studio project, the Content Pipeline integrates it into the Visual Studio build just as it would any other source file, providing error handling, status information, and other standard build features. For information on how to set Content Pipeline build options, see [Game Asset Properties](#).

In the course of a build, the Content Pipeline invokes four principal components to perform different parts of the transformation from a DCC output file into a binary part of an XNA Game Studio game.

1. [Importer](#)
2. [Content Processor](#)
3. [Content Compiler](#)
4. [Content Loader](#)

The following figure shows the flow of this build process.



Importer and Content DOM Types

XNA Game Studio provides a number of standard importers, which are listed in [Standard Importers and Processors](#). This includes an importer for the Autodesk .fbx format, and one for the DirectX .x format. These importers simplify the importing of art assets, since many DCC tools can export content to one of these formats as well as to their own native formats.

For art assets that are available only in formats not supported by XNA Game Studio standard importers, custom importers may be available as well. Such custom importers can be developed by DCC vendors, game-engine developers, or interested game hobbyists. For more information about how to do this, see [How To: Write a Custom Importer and Processor](#). Once you install a custom importer on your computer, you can associate those art files with the importer in order to invoke the importer whenever you build the art files (see [Using a Custom Importer or Content Processor](#)).

In many cases, Content Pipeline importers convert any content they can into managed objects based on the Content Document Object Model (DOM), which includes strong typing for assets such as meshes, vertices, and materials.

The Content Pipeline can use XML cache files in subsequent passes to speed up game content builds as well as to debug. When a content processor requests that a specified file be imported (typically using the [BuildAndLoadAsset](#) method in its [Process](#) function) and there is an up-to-date cache file already, the Content Pipeline deserializes the cache file instead of invoking the importer. These XML cache files are not used externally, however, because their format may well change in future releases.

Instead of producing standard Content DOM objects, a custom importer may produce custom objects for a particular custom content processor to consume.

Content Processor

A content processor accepts as input the output generated by an importer. Each content processor is tied to specific object types. For instance, the [Effect Processor](#) accepts only [EffectContent](#) objects, representing a DirectX Effect asset. As discussed previously, in many cases, this output consists of standard Content DOM objects, but may also consist of custom objects.

A content processor then produces managed objects that can be used in a game at run time. In the case of standard Content DOM objects, this transformation can be performed by classes in the Content Pipeline class library. However, if a content processor generates custom managed objects, you must provide full functionality for them, including saving and loading to and from a binary file. For more information, see [How To: Write a Custom Importer and Processor](#).

Content Compiler

After you add the various game assets to the project and the content processors generate managed code, the managed code is serialized into a compact binary format (also referred to as an intermediate format) by the Content Pipeline content compiler. This format is tightly coupled to the XNA Framework. It is not designed for use by other run-time libraries. At this point, the asset has been processed by the Content Pipeline and is in a format that can be used by your game at runtime. See the diagram above for details.

Content Loader

When you need the compiled asset in a game, call the [ContentManager.Load](#) method to invoke the content loader. The content loader then locates and loads the asset into the memory space of the game where you can access it.

See Also [Content Pipeline Overview of the Content Pipeline](#)

Content Compression

XNA Game Studio offers data compression - an easy way to decrease the size of certain built game assets (e.g. textures, shaders, and meshes). This reduces the size required for media storage and downloads, and it reduces the deploy time required when debugging a game.

Transparent Compression

Compression and decompression of game assets is a transparent operation to games. You don't need to modify your game to make use of compressed data.

- Compression occurs as part of the content pipeline portion of the build process. It is performed automatically when you select the property in the [Content Build](#) page of the Project Designer.
- Decompression occurs automatically when you access the assets through [ContentManager.Load](#).

Note

Compression of game assets is not available for Zune projects. Microsoft has concluded that the small benefit gained by compressing data for Zune is exceeded by the performance requirements of the Zune processor for decompression.

Compression Algorithm

The compression used by XNA Game Studio uses a variant of the LZ family of compression algorithms, which is an efficient, lossless method.

The ratio of compressed data to source data can vary significantly depending on the character of the source data. However, the average compression ratio for most game asset types is around 60 percent.

The automatic compression algorithms will not compress source data that is very small (a file size less than one disk sector), as the performance benefits of compression are typically lost or even made worse by the execution overhead to decompress the data. These assets will be saved in uncompressed format.

Disabling Compression

Some content types, such as [Song Class](#) and [SoundEffect Class](#) data, may not benefit from a general lossless compression algorithm. This is especially true if the data is already compressed in a specialized format.

When implementing a custom content importer, compression can be disabled by overriding the method [ContentTypeWriter.ShouldCompressContent](#), which must return **false**.

If compression is to be permitted, no action is required. The base class definition of [ContentTypeWriter.ShouldCompressContent](#) returns **true**.

See Also [Content Pipeline](#)

Standard Importers and Processors

Describes the standard Content Pipeline importers and content processors of XNA Game Studio that support various common art-asset file formats.


Importers and content processors are implemented as assemblies. In addition to the standard ones provided by XNA Game Studio and listed below, you can also use custom importers and processors that you or other third parties develop. Use the Properties window to assign an appropriate importer and processor for each game asset you add to your game project (see [Game Asset Properties](#) for more information).

Standard Importers

The table below describes the standard importers shipped with XNA Game Studio and the type of game asset each supports.

All standard importers are declared as part of the Microsoft.Xna.Framework.Content.Pipeline namespace.

Name	Type Name	Output Type	Description
Autodesk FBX - XNA Framework	FbxImporter	NodeContent	Imports game assets specified with the Autodesk FBX format (.fbx). This importer is designed to work with assets exported with the 2006.11 version of the FBX exporter.
Effect - XNA Framework	EffectImporter	EffectContent	Imports a game asset specified with the DirectX Effect file format (.fx).
Sprite Font Description - XNA Framework	FontDescriptionImporter	FontContent	Imports a font description specified in a .spritefont file.
Texture - XNA Framework	TextureImporter	TextureContent	Imports a texture. The following types are supported: .bmp, .dds, .dib, .hdr, .jpg, .pfm, .png, .ppm, and .tga.
X File - XNA Framework	XImporter	NodeContent	Imports game assets specified with the DirectX X file format (.x). This importer expects the coordinate system to be left-handed.

XACT Project - XNA Framework	N/A	N/A	<p>Imports game audio specified in the Microsoft Cross-Platform Audio Creation Tool (XACT) format (.xap).</p> <p> Tip Associating an .xap file with XACT allows you to automatically open XACT when editing any .xap file. Associate the file in XNA Game Studio by right-clicking the .xap file in Solution Explorer, and clicking Open With. Once the dialog box is open, select the XACT-specific string, and then click Set As Default. If the XACT-specific option is not available, you must run XACT before this option appears. The application is available from the Microsoft XNA Game Studio group. From the Start Menu, click Tools, and then click Microsoft Cross-Platform Audio Creation Tool (XACT).</p>
XML Content - XNA Framework	Xml Importer	object	<p>Imports XML content used for editing the values of a custom object at runtime. For instance, you could pass XML code to this importer that looks for the specified property of a custom type and changes it to the specified value. You could then process the custom object with a processor or pass it to your game untouched using the No Processing Required processor.</p> <p>This importer is designed for scenarios like importing an XML file that describes game data at runtime (similar to the Sprite Font Description importer) or importing terrain data in an XML file that is then passed to a processor that generates a random terrain grid using that data.</p>

Standard Content Processors

XNA Game Studio ships with a variety of processors that support several common game asset types. Many of the standard processors, such as the [TextureProcessor](#), support parameters for modifying the default behavior of the processor. For more information, see [Parameterized Processors](#).

The following table describes the standard processors and the type of game asset each supports.

Name	Type Name	Input Type	Output Type	Description
Effect - XNA Framework	EffectProcessor	EffectContent	CompiledEffect	Compiles the string in EffectContent to the appropriate platform.
Model - XNA Framework	ModelProcessor	NodeContentClass	ModelContentClass	<p>A parameterized processor that outputs models as a ModelContent Class object.</p> <p>Available parameters:</p> <ul style="list-style-type: none"> • Color Key Color - Any valid Color. Magenta is the default value. • Color Key Enabled - A Boolean value indicating if color keying is enabled. The default value is true. • Generate Mipmaps - A Boolean value indicating if mipmaps are generated. The default value is false. • Generate Tangent Frames - A Boolean value indicating if tangent frames are generated. The default value is false. • Resize Textures to Power of Two - A Boolean value indicating if a texture is resized to the next largest power of 2. The default value is false. • Scale - Any valid float value. The default value is 1.0. • Swap Winding Order - A Boolean value indicating if the winding order is swapped. This is useful for models that appear to be drawn inside out. The default value is false. • Texture Format - Any valid SurfaceFormat value. Textures are either unchanged, converted to the Color format, or DXT Compressed. For more information, see TextureProcessorOutputFormat. • X Axis Rotation - Amount, in degrees of rotation. The default value is 0. • Y Axis Rotation - Amount, in degrees of rotation. The default value is 0. • Z Axis Rotation - Amount, in degrees of rotation. The default value is 0.

No Processing Required	PassThroughProcessor	Object	Object	<p>Performs no processing on the file.</p> <p>Select this processor if your content is already in a game-ready format (for example, an externally prepared DDS file) or a specialized XML format (.xml) designed for use with XNA Game Studio.</p>
Sprite Font Description - XNA Framework	FontDescriptionProcessor	FontDescription	SpriteFontContent	Converts a .spritefont file specifying a font description into a font.
Sprite Font Texture - XNA Framework	FontTextureProcessor	TextureContent	SpriteFontContent	<p>A parameterized processor that outputs a sprite font texture as a SpriteFontContent object.</p> <p>Available parameters:</p> <ul style="list-style-type: none"> • First Character - Any valid character. The space character is the default value.
Sprite Font Texture - XNA Framework	FontTextureProcessor	Texture2DContent	SpriteFontContent	Converts a specially marked 2D bitmap file (.bmp) into a font. Pixels of Color.Magenta are converted to Color.TransparentBlack .
Texture - XNA Framework	TextureProcessor	TextureContentClass	TextureContentClass	<p>A parameterized processor that outputs textures as a TextureContentClass object.</p> <p>Available parameters:</p> <ul style="list-style-type: none"> • Color Key Color - Any valid Color. Magenta is the default value. • Color Key Enabled - A Boolean value indicating if color keying is enabled. The default value is true. • Generate Mipmaps - A Boolean value indicating if mipmaps are generated. The default value is false. • Resize to Power of Two - A Boolean value indicating if a texture is resized to the next largest power of 2. The default value is false. • Texture Format - Any valid SurfaceFormat value. Textures are either unchanged, converted to the Color format, or DXT Compressed. For more information, see TextureProcessorOutputFormat.
XACT Project - XNA Framework	N/A	N/A	N/A	Generates audio assets from an XACT project.

See Also [Content Pipeline](#)

[Overview of the Content Pipeline](#)

[How To: Write a Custom Importer and Processor](#)

[Using a Custom Importer or Content Processor](#)

Parameterized Processors

Describes how parameterized processors work in XNA Game Studio. Many of the standard Content Pipeline content processors shipped with XNA Game Studio support parameter usage. Parameterization makes any standard or custom processor more flexible and better able to meet the needs of your XNA Framework application. In addition to specifying values for standard parameters, you can easily implement parameter support for a new or existing custom processor. For more information, see [Developing with Parameterized Processors](#).

When you select a game asset, the Properties window displays the parameters for the related asset processor. Use the Properties window at any time to modify these parameter values.

Note

If you change the processor for a game asset to a different processor, all parameter values are reset to their default values. This means that if you modify the **Generate Mipmaps** parameter value for the [TextureProcessor](#), then switch to a different processor (for example, [FontTextureProcessor Class](#)), the parameters would be switched to the default values for that processor. If you then switch back again, the modified values are reset to the default values of the original processor. *The values do not revert to the modified values you set originally.*

Standard Parameterized Processors

The following table describes only standard processors that accept parameters, the parameter types, and their default value. For more information on all standard processors, see [Standard Importers and Processors](#).

Friendly Name	Type Name	Input Type	Output Type	Description
Model - XNA Framework	ModelProcessor	NodeContent Class	ModelContent Class	<p>A parameterized processor that outputs models as a ModelContent Class object.</p> <p>Available parameters:</p> <ul style="list-style-type: none"> • Color Key Color—Any valid Color. Magenta is the default value. • Color Key Enabled—A Boolean value indicating if color keying is enabled. The default value is true. • Generate Mipmaps—A Boolean value indicating if mipmaps are generated. The default value is false. • Generate Tangent Frames—A Boolean value indicating if tangent frames are generated. The default value is false. • Resize Textures to Power of Two—A Boolean value indicating if a texture is resized to the next largest power of 2. The default value is false. • Scale—Any valid float value. The default value is 1.0. • Swap Winding Order—A Boolean value indicating if the winding order is swapped. This is useful for models that appear to be drawn inside out. The default value is false. • Texture Format—Any valid value from TextureProcessorOutputFormat. Textures are either unchanged, converted to the Color format, or DXT Compressed. • X Axis Rotation—Amount, in degrees of rotation. The default value is 0. • Y Axis Rotation—Amount, in degrees of rotation. The default value is 0. • Z Axis Rotation—Amount, in degrees of rotation. The default value is 0.
Sprite Font Texture - XNA Framework	FontTextureProcessor	TextureContent Class	SpriteFontContent	<p>A parameterized processor that outputs a sprite font texture as a SpriteFontContent object.</p> <p>Available parameters:</p> <ul style="list-style-type: none"> • First Character—Any valid character. The space character is the default value.

Texture - XNA Framework	TextureProcessor	TextureContentClass	TextureContentClass	<p>A parameterized processor that outputs textures as a TextureContent Class object.</p> <p>Available parameters:</p> <ul style="list-style-type: none"> • Color Key Color—Any valid Color. Magenta is the default value. • Color Key Enabled—A Boolean value indicating if color keying is enabled. The default value is true. • Generate Mipmaps—A Boolean value indicating if mipmaps are generated. The default value is false. • Resize to Power of Two—A Boolean value indicating if a texture is resized to the next largest power of 2. The default value is false. • Texture Format—Any valid value from TextureProcessorOutputFormat. Textures are either unchanged, converted to the Color format, or DXT Compressed.
-------------------------	------------------	---------------------	---------------------	--

Texture Processing with XNA Game Studio

In versions before XNA Game Studio 2.0, the following standard processors were used for texture processing.

- TextureProcessor
- ModelTextureProcessor
- SpriteTextureProcessor

In XNA Game Studio, the Texture - XNA Framework processor replaces the functionality of this entire group. For new XNA Game Studio projects and projects converted by the Project Upgrade Wizard for XNA Game Studio 2.0, the correct processor is selected automatically. However, for existing custom processors that make calls to any processor from the list above, the code must be modified to use the new [TextureProcessor](#) with the proper parameters. The following table describes the parameter values required for emulating each previous texture processor.

Processor Type Name	Required Parameters and Values
TextureProcessor	<p>Call the Texture—XNA Framework processor (TextureProcessor) with the following parameter values:</p> <ul style="list-style-type: none"> • GenerateMipmaps: false • Destination format: NoChange • Colorkey Enabled: false • Colorkey Value: Magenta • Resize: false
ModelTextureProcessor	<p>Call the Texture—XNA Framework processor (TextureProcessor) with the following parameter values:</p> <ul style="list-style-type: none"> • GenerateMipmaps: true • Destination format: DXTCompressed • Colorkey Enabled: true • Colorkey Value: Magenta • Resize: false
SpriteTextureProcessor	<p>Call the Texture—XNA Framework processor (TextureProcessor) with the following parameter values:</p> <ul style="list-style-type: none"> • GenerateMipmaps: false • Destination format: Color • Colorkey Enabled: true • Colorkey Value: Magenta

For more information on modifying and declaring parameter values programmatically, see [Developing with Parameterized Processors](#).

See Also [Content Pipeline](#)

Developing with Parameterized Processors

Describes how developing with parameterized processors, both standard and custom, requires additional thought and care. This topic discusses a method for programmatically modifying existing parameter values, and adding new parameters to your own processors.

Programmatically Setting Parameter Values

When you need to pass parameter values from one processor to another (also referred to as chaining), use the [BuildAsset](#) and [BuildAndLoadAsset](#) methods. Pass the parameter and its value using the *ProcessorParameters* argument of the respective function. For example, a custom model processor would invoke a second processor for model textures with a call to [BuildAsset](#) and pass any parameter values in the *ProcessorParameters* argument.

The following code example demonstrates this technique. First, add several parameters to a data dictionary:

```
//create a dictionary to hold the processor parameter
OpaqueDataDictionary parameters = new OpaqueDataDictionary();

//add several parameters to the dictionary
parameters.Add( "ColorKeyColor", Color.Magenta );
parameters.Add( "ColorKeyEnabled", true );
parameters.Add( "ResizeToPowerOfTwo", true );
```

After adding the necessary parameters, pass the dictionary to the chained processor:

```
context.BuildAsset<TextureContent, TextureContent="">(
    texture, typeof( TextureProcessor ).Name,
    parameters,
    null,
    null );
```

This call passes all parameters (stored in *parameters*) to a texture processor.

As stated earlier, any parameters not recognized by the receiving processor are ignored. Therefore, if the parameter *ColorKeyCode* was entered into the dictionary as *ColourKeyCode*, it would be ignored by the receiving processor.

Declaring Process Parameters

Adding one or more parameters to your custom processor requires additional code in your processor's definition. Parameters support the following types:

- bool
- byte
- sbyte
- char
- decimal
- double
- float
- int
- uint
- long
- ulong
- short
- ushort
- string
- enum
- [Vector2](#), [Vector3](#), and [Vector4](#)
- [Color](#)

Parameters of other types are ignored by the processor.

Tip

Apply the `Browsable` attribute (with a value of **false**) to an individual parameter to prevent that parameter from being displayed in the Properties window.

The following code example defines a simple custom processor that switches the coordinate system of a model using a single parameter (called `switchCoordinateSystem`):

```
public class SwitchCoordSystemProcessor : ModelProcessor
{
    #region Processor Parameters
    private bool switchCoordinateSystem = false;

    [DisplayName("Switch Coordinate System")]
    [DefaultValue(false)]
    [Description("Switches the coordinate system of a model.")]
    public bool SwitchCoordinateSystem
    {
        get { return switchCoordinateSystem; }
        set { switchCoordinateSystem = value; }
    }
}
//additional class code follows...
```

In this code, the `SwitchCoordSystemProcessor` class is derived from `ModelProcessor`. This indicates that the processor accepts a model as input. The next few lines declare a single property called `SwitchCoordinateSystem` of type **bool**. Please note that every parameter must have a **set** method. The property also has several attributes applied to it:

Attribute Name	Usage
DisplayName	Name of the property when it appears in the Properties window of XNA Game Studio. If not specified, the internal property name, declared in the source code, is used. For this example, "Switch Coordinate System" would be displayed.
DefaultValue	A UI hint specifying the default value the property could have. This value is used only as a UI hint. It will not be set on the property or override the default value declared in the code.
Description	Descriptive text displayed when you select the property in the Properties window of XNA Game Studio.

This completes the definition of the `SwitchCoordinateSystem` property.

In the next code example, the class definition is continued with an override of the `Process` method:

```
//additional class code precedes...

public override ModelContent Process(NodeContent input, ContentProcessorContext context)
{
    if (switchCoordinateSystem)
    {
        Matrix switchMatrix = Matrix.Identity;
        switchMatrix.Forward = Vector3.Backward;
        MeshHelper.TransformScene(input, switchMatrix);
    }

    return base.Process(input, context);
}
```

This code passes the `SwitchCoordinateSystem` property (declared earlier) value to `TransformScene`, which is a helper method that applies a transform to a scene hierarchy.

See Also [Content Pipeline Parameterized Processors](#)

Using a Custom Importer or Content Processor

Describes how to use a custom processor or importer in an existing game solution. XNA Game Studio provides standard importers and processors for a number of common file formats used to store basic game assets such as models, materials, effects, sprites, textures, and so on.

Note

For a list of the file formats that these standard importers and processors support, see [Standard Importers and Processors](#).

If you have game assets saved in a format that the standard importers and processors do not support, you may be able to find a custom third-party importer or a processor for XNA Game Studio that supports that file format. If you have enough information about the file format, you can even build your own custom importer or processor, as described in [How To: Write a Custom Importer and Processor](#).

Security Note

Before you open an existing project or component, determine the trustworthiness of the code outside of the Visual Studio designer. Opening projects or components in the Visual Studio designer automatically executes that code on your local machine in the trusted process of VCSEXPRESS.exe or DevEnv.exe.

The following procedure shows how to add such a custom importer and processor to an existing game project. These steps assume that you have copied the new importer or processor to a local subfolder of the game project in question.

To Add a Custom Importer or Processor to a Game Project

1. Open XNA Game Studio.
2. Load the solution associated with your game.
3. From Solution Explorer, right-click the nested content project node, and click **Add Reference**.
4. Navigate to the directory containing the assembly with the custom importer or processor, and then add it to the solution.
5. Save the solution.

The new importer or processor now appears as one of the available choices for importing or processing a newly added game asset.

See Also [Content Pipeline](#)
[Standard Importers and Processors](#)

Extending an XNA Framework Standard Processor

Describes how XNA Game Studio lets you modify or extend the behavior of any of the standard Content Pipeline processors that ship with the product. See [Standard Importers and Processors](#) for a description of the standard processors.

Because there are so many asset variants supported by different digital content creation (DCC) tools, it is often useful to be able to modify how one of the standard processors operates. The following examples illustrate some of the kinds of things you might want to do.

Note

The following code samples are for demonstration purpose only. Most of the functionality described is already available by using parameters on a standard processor.

Adding a Scaling Operation to a Processor

There are many reasons why you might want to modify the existing functionality of a standard processor. Here is one example. If your source assets and your game are at different scales, you might want the processor to scale each model automatically at build time. You can implement such automatic scaling by overriding the [Process](#) method of the [ModelProcessor](#) class, which generates a [Model](#). In the override, you would first scale the entire scene and then invoke the base class functionality to process as usual.

The following code illustrates this technique:

```
[ContentProcessor]
class ScalingModelProcessor : ModelProcessor
{
    public override ModelContent Process(
        NodeContent input, ContentProcessorContext context )
    {
        MeshHelper.TransformScene( input, Matrix.CreateScale( 10.0f ) );
        return base.Process( input, context );
    }
}
```

Generating Additional Data

In some cases, you might want to add information to a game asset that a standard processor would not. For example, if a custom effect you want to apply requires tangent or binormal data, you can extend the standard model processor to build this additional data into the asset. To do this, you would override the [Process](#) method of the [ModelProcessor](#) class. In the override, navigate the [NodeContent](#) hierarchy of the game asset, and call [CalculateTangentFrames](#) for each [MeshContent](#) object you find.

The following code shows how you would do this:

```
[ContentProcessor]
class ModelProcessorWithTangents : ModelProcessor
{
    public override ModelContent Process( NodeContent input, ContentProcessorContext context )
    {
        GenerateTangentFramesRecursive( input );
        return base.Process( input, context );
    }

    private void GenerateTangentFramesRecursive( NodeContent node )
    {
        MeshContent mesh = node as MeshContent;
        if (mesh != null)
        {
            MeshHelper.CalculateTangentFrames( mesh, VertexChannelNames.TextureCoordinate( 0 ),
                VertexChannelNames.Tangent( 0 ), VertexChannelNames.Binormal( 0 ) );
        }

        foreach (NodeContent child in node.Children)
        {

```



```

        GenerateTangentFramesRecursive( child );
    }
}
}

```

Changing the Processors Called for Child Objects

Another technique that can be useful is to override a standard processor and change the way child objects are processed by changing the processors that are used for them.

Consider, for example, the hierarchy of calls through which textures in a model are processed:

- The standard `ModelProcessor.Process` method is called to process a `NodeContent` object that represents the root of a scene.
- `ModelProcessor.Process` in turn calls the `ModelProcessor.ConvertMaterial` method once for every `MaterialContent` object used in the scene.
- `ModelProcessor.ConvertMaterial` in turn invokes the `MaterialProcessor.Process` method on the `MaterialContent` object passed to it.
- `MaterialProcessor.Process` in turn calls the `MaterialProcessor.BuildTexture` method once for each texture in the `MaterialContent.Textures` collection in the `MaterialContent` object passed to it.
- `MaterialProcessor.BuildTexture` in turn invokes the `ModelTextureProcessor.Process` method on the `TextureContent` object passed to it.

One reason that you might want to change how this works is that the `ModelTextureProcessor.Process` method applies DXT1 or DXT5 compression to all textures it processes. If textures in your game assets are compressed already, you might well wish to avoid a second compression.

Here is how to prevent compression from being applied to model textures during processing:

1. Create an override of the standard `MaterialProcessor.BuildTexture` method, and invoke the `TextureProcessor.Process` method, which does no compression, instead of `ModelTextureProcessor.Process`.
2. Create an override of `ModelProcessor.ConvertMaterial` that invokes your override of `MaterialProcessor.BuildTexture` instead of the standard one.

The first of these overrides could be coded as follows:

```

[ContentProcessor]
class NoCompressionMaterialProcessor : MaterialProcessor
{
    protected override ExternalReference<TextureContent> BuildTexture(
        string textureName, ExternalReference<TextureContent> texture, ContentProcessor
Context context )
    {
        return context.BuildAsset<TextureContent, TextureContent>( texture, "TexturePro
cessor" );
    }
}

```

There are several things to note about this code:

- An `ExternalReference` is an asset object that is shared between multiple classes, such as a diffuse texture used by more than one material. When such an asset is specified, the Content Manager loads only one copy of the `ExternalReference` at run time and builds it only once, no matter how many references there are to it.
- The `ContentProcessorContext.BuildAsset` method lets you invoke a processor by name to build the content in an object.
- Although `textureName`, the first argument to `BuildTexture`, is ignored in the override above, you could use it if you wanted to process textures differently depending on normal maps or other criteria.

Given the processor created by your first override above, you could code the second override as follows:

```

[ContentProcessor]
class NoCompressionModelProcessor : ModelProcessor
{
    protected override MaterialContent ConvertMaterial(
        MaterialContent material, ContentProcessorContext context )
    {
        return context.Convert<MaterialContent, MaterialContent>(

```

```
        material, "NoCompressionMaterialProcessor" );  
    }  
}
```

Because this override is processing `MaterialContent` objects in memory rather than `ExternalReference` objects, it uses the `ContentProcessorContext.Convert` function instead of `BuildAsset` to invoke the processor created by your first override.

After building and installing your new `NoCompressionModelProcessor` (see [Using a Custom Importer or Content Processor](#)), you can assign it to any models whose textures are already compressed and no further compression will be applied to them.

See Also

Concepts

[Overview of the Content Pipeline](#)

[Content Pipeline Architecture](#)

[Standard Importers and Processors](#)

[Using a Custom Importer or Content Processor](#)

[Content Pipeline Content Catalog at XNA Creators Club Online](#)

Sprite Font XML Schema Reference

Describes the valid tags and values for Sprite-Font (.spritefont) XML files used by the Content Pipeline to create [SpriteFont](#) textures.

Tag Name	Content Type	Content Description
<FontName>	string	The name of the font to be imported. This is not the name of a font file, but rather the friendly name that identifies the font once it is installed on your computer. You can use the Fonts folder in Control Panel to see the names of fonts installed on your system, and to install new ones as well. The Content Pipeline supports the same fonts as the System.Drawing.Font class, including TrueType fonts but not bitmap (.fon) fonts.
<Size>	float	The point size of the font to be imported.
<Spacing>	float	The number of pixels to add between each character when the string is rendered.
<UseKerning>	Boolean	Specifies whether to use kerning information when rendering the font. Default value is true .
<Style>	"Regular," "Bold," "Italic," or "Bold, Italic"	The style of the font to be imported.
<DefaultCharacter>	char	The Unicode character to substitute any time an attempt is made to render characters that are not in the font. Specifying this element is optional.
<CharacterRegions>	One or more <CharacterRegion> tags	One or more numerical ranges indicating which subset of Unicode characters to import.
<CharacterRegion>	One <Start> and one <End> tag	The beginning and end of a region of Unicode characters.
<Start>	char	The first Unicode character to include in a <CharacterRegion>.
<End>	char	The last Unicode character to include in a <CharacterRegion>.

Example

Here is a sample .spritefont file:

```
<?xml version="1.0" encoding="utf-8"?>

<XnaContent xmlns:Graphics="Microsoft.Xna.Framework.Content.Pipeline.Graphics">
  <Asset Type="Graphics:FontDescription">
    <FontName>Courier New</FontName>
    <Size>18</Size>
    <Spacing>0</Spacing>
    <UseKerning>true</UseKerning>
    <Style>Regular</Style>
    <CharacterRegions>
      <CharacterRegion>
        <Start>32</Start>
        <End>127</End>
      </CharacterRegion>
    </CharacterRegions>
  </Asset>
</XnaContent>
```

See Also
Concepts

[2D Graphics Overview](#)

Tasks

[How To: Draw Text](#)

Reference

[SpriteFont](#)

How To: Write a Custom Importer and Processor

Describes how a content importer adds support for a new art asset file format.

You will need to write a new content importer to add support. Also, you may need to write a custom processor, writer, and reader for the new art asset type after it has been imported.

The Complete Sample

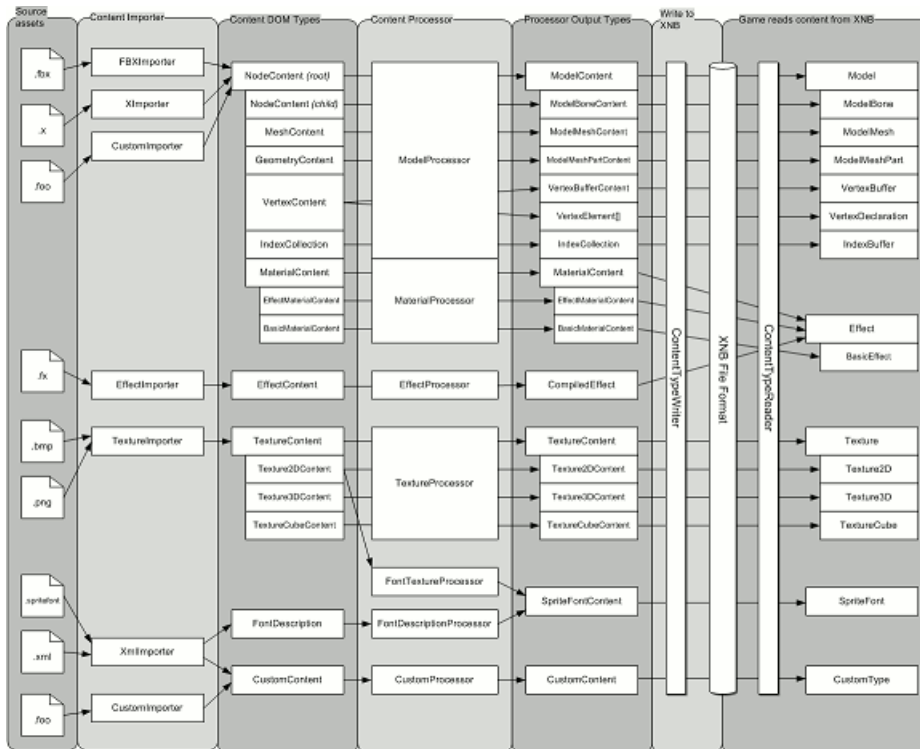
The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download CPExtPixelShader_Sample.zip.](#)

Overview

XNA Game Studio already provides standard content pipeline importers and processors to support common art-asset file formats, as described in [Standard Importers and Processors](#). Third parties also provide custom importers and processors to support additional formats. Currently, the XNA Game Studio Content Document Object Model (DOM) provides support for meshes, materials, textures, sprite-fonts, and animations. Outside of these, a custom importer can return a [ContentItem](#) with custom information in its opaque data, or a custom type you have developed.

The following diagram lists the complete Content DOM.



However, if you want to support a new type in the content pipeline, writing your own importer and processor can be fairly straightforward.

For example, suppose you want to compile HLSL source files into pixel shaders. You want the result to be somewhat like the [EffectImporter](#) and [EffectProcessor](#) classes built into XNA Game Studio, but you want to process individual pixel shaders rather than complete effects. This topic provides a simple example to show you the steps you take to write an importer and processor, and also the writer and reader you need to save and load the results. The sections below describe each of the steps.

- [Creating a Content Pipeline Extension Library](#)
- [Implementing a Simple Importer](#)
- [Implementing a Processor to Compile the Shader](#)
- [Implementing a Writer for the Compiled Shader](#)
- [Implementing a Reader for the Pixel Shaders](#)
- [Using the Output of the New Processor in Your Game](#)

Creating a Content Pipeline Extension Library

The first step in writing an importer and processor is to create a new project for them. You need to do this because your importer and processor are used by the content pipeline when your game is being built. They are not part of the game itself. As a result, you need to provide them as a separate library assembly that the content pipeline can link to when it needs to build the new file format you are supporting.

To create a content pipeline extension library

1. In XNA Game Studio, load a game solution you are developing (the sample uses "CPExtPixelShader").
2. From Solution Explorer, right-click the **Solution** node, click **Add**, and then click **New Project**.
3. From the **Add New Project** dialog box, under the **Visual C#** node, from the Project types: pane, select the **XNA Game Studio 3.1** node.
4. Select the **Content Pipeline Extension Library (3.1)** template, assign a name to the new project at the bottom of the dialog box (name this project `PSProcessorLib`), and click **OK**.
5. From Solution Explorer, right-click the **ContentProcessor1.cs** item, and click **Delete**.

The remaining steps create a reference to the `PSProcessorLib` content extension project.

6. From Solution Explorer, right-click the **Content** node of the `CPExtPixelShader` project, and then click **Add Reference**.
7. From the **Projects** tab, select your content extension project, and click **OK**.

The new project is now ready for your custom importer and processor implementation.

Implementing a Simple Importer

Follow these steps to add a content importer to your processor.

To implement a simple importer

1. Create a class to hold the input data you are importing.

In this case, it takes the form of a string of HLSL source code.

2. Add a new C# class named **PSSourceCode** to the processor project.

The first thing to do in the file containing your new class definition is add the following **using** statement at the beginning of the file:

C#

```
using Microsoft.Xna.Framework.Content.Pipeline;
```

3. Now define the class as follows:

C#

```
class PSSourceCode
{
    public PSSourceCode(string sourceCode)
    {
        this.sourceCode = sourceCode;
    }

    private string sourceCode;
    public string SourceCode { get { return sourceCode; } }
}
```

4. Write an importer class to import the HLSL source code.

This class must be derived from `ContentImporter` and implement the `Import` method. All it does is read a text file containing HLSL source code into your **PSSourceCode** class.

5. Using the **New Item** dialog box, add a new **Content Importer** item (called **PSImporter**) to the processor project.

6. Now define the class as follows:

C#

```
[ContentImporter(".psh", DefaultProcessor = "PSProcessor",
    DisplayName = "Pixel Shader Importer")]
class PSImporter : ContentImporter<PSSourceCode>
{
    public override PSSourceCode Import(string filename,
        ContentImporterContext context)
    {
        string sourceCode = System.IO.File.ReadAllText(filename);
        return new PSSourceCode(sourceCode);
    }
}
```

The [ContentImporter](#) attribute applied to the **PSImporter** class provides some context for the user interface of XNA Game Studio. Since this importer supports files with a .psh extension, XNA Game Studio automatically selects the **PSImporter** importer when a .psh file is added to the project. In addition, the *DefaultProcessor* argument specifies which processor XNA Game Studio selects when a .psh file is added.

Note

To specify multiple file types, separate with a comma the file extensions listed in the [ContentImporterAttribute](#). For example, `[ContentImporter (".bmp", ".dds", ".tga")]` declares an importer that accepts .bmp, .dds, and .tga file types. Normally, an importer that accepts multiple file formats is specialized to generate one particular kind of output type, such as textures. However, aside from difficulties of maintenance, there is nothing to prevent a single importer from being written to handle many different content types.

When the game is built, the [ContentImporter.Import](#) function is called once for each XNA content item in the current project.

When invoked against an input file in the appropriate format, a custom importer is expected to parse the file and produce as output one or more content objects of appropriate types. Since an importer's output is passed directly to a content pipeline processor, each type that an importer generates must have at least one processor available that can accept it as input.

Tip

An importer that generates DOM objects may also automatically generate an intermediate XML cache file that serializes these objects. For this to happen, the importer must be implemented with the [CacheImportedData](#) attribute flag set to **true**. This flag is **false** by default. To set the attribute flag to **true**, begin the implementation of your Importer class like this:

```
[ContentImporter( ".MyExt", CacheImportedData = true )]
class PSImporter : ContentImporter<PSSourceCode>
{
    ...
}
```

Implementing a Processor to Compile the Shader

After the new importer has read in the pixel shader source code from a text file, your content processor takes over and compiles the shader into binary form.

To write the processor

1. Create a class to store the compiled output, which in this case takes the form of an array of bytes.
2. Add a C# class called **CompiledPS** to the processor project, and define the new class as follows:

C#

```
class CompiledPS
{
    public CompiledPS(byte[] compiledShader)
    {
        this.compiledShader = compiledShader;
    }
}
```

```

    }

    private byte[] compiledShader;
    public byte[] CompiledShader {
        get { return (byte[])compiledShader.Clone(); }
    }
}

```

Now you are ready to write the processor class, which converts a **PSSourceCode** object into a **CompiledPS** object.

- Using the **New Item** dialog box, add a new **Content Processor** item (called **PSProcessor**) to the processor project.
- Now define the class as follows:

C#

```

[ContentProcessor(DisplayName = "Pixel Shader Processor")]
class PSProcessor : ContentProcessor<PSSourceCode, CompiledPS>
{
    public override CompiledPS Process(PSSourceCode input,
        ContentProcessorContext context)
    {
        CompiledShader shader =
            ShaderCompiler.CompileFromSource(input.SourceCode, null, null,
            CompilerOptions.None, "main",
            ShaderProfile.PS_2_0, context.TargetPlatform);
        if (!shader.Success)
        {
            throw new InvalidContentException(shader.ErrorsAndWarnings);
        }
        return new CompiledPS(shader.GetShaderCode());
    }
}

```

The [Framework.Graphics.ShaderCompiler](#) class compiles the shader to binary code that runs on the platform targeted by your game. The [context.TargetPlatform](#) argument targets the platform. If an error occurs during compilation, **PSProcessor** throws an [InvalidContentException](#). The error appears in the **Error List** window of XNA Game Studio.

Implementing a Writer for the Compiled Shader

The final design-time class to implement is a writer that saves the compiled pixel shader produced by your processor as a binary .xnb file.

To implement the writer for the compiled shader

- Using the **New Item** dialog box, add a new **Content Type Writer** item (called **PSWriter**) to the processor project.
- Define the new class as follows:

C#

```

[ContentTypeWriter]
class PSWriter : ContentTypeWriter<CompiledPS>
{
    protected override void Write(ContentWriter output, CompiledPS value)
    {
        output.Write(value.CompiledShader.Length);
        output.Write(value.CompiledShader);
    }
    public override string GetRuntimeType(TargetPlatform targetPlatform)
    {
        return typeof(PixelShader).AssemblyQualifiedName;
    }
}

```



```

public override string GetRuntimeReader(TargetPlatform targetPlatform)
{
    return "CPEXPixelShader.PSReader, CPEXPixelShader," +
        " Version=1.0.0.0, Culture=neutral";
}
}

```

The `GetRuntimeType` method identifies the type of object your game should load from the .xnb file written by the writer object. In this instance, the .xnb file contains the binary array from your custom **CompiledPS** type, and this method identifies how that array will be mapped to a standard `Framework.Graphics.PixelShader` object type at load time.

The `GetRuntimeReader` method specifies what reader should be invoked to load the .xnb file in your game. It returns the namespace and name of the reader class, followed by the name of the assembly in which that class is physically located.

3. In your code, change the assembly name to match the actual name of your game and its assembly, since that is where you will be loading the shaders.

At this point, the code for your **PSProcessorLib** library is complete.

Implementing a Reader for the Pixel Shaders

Now move from the **PSProcessorLib** library project back to your game project and write the class that your game uses to load the .xnb files that your processor creates. This is the class that your writer specified previously as its reader.

To implement a reader for the pixel shaders

1. In your game project, add a C# class called **PSReader** to your game project.
2. Add the **using** statements you will need at the top of the file:

C#

```

using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.Graphics;

```

3. Deriving from the `ContentTypeReader` generic class for the `PixelShader` type, override the **Read** method, and define your class as follows:

C#

```

class PSReader : ContentTypeReader<PixelShader>
{
    /// <summary>
    /// Loads an imported shader.
    /// </summary>
    protected override PixelShader Read(ContentReader input,
        PixelShader existingInstance)
    {
        int codeSize = input.ReadInt32();
        byte[] shaderCode = input.ReadBytes(codeSize);

        IGraphicsDeviceService graphicsDeviceService =
            (IGraphicsDeviceService)input.ContentManager.ServiceProvider.
                GetService(typeof(IGraphicsDeviceService));
        return new PixelShader(graphicsDeviceService.GraphicsDevice,
            shaderCode);
    }
}

```

4. At this point, build the processor project.

Once it has completed, you are ready to use the new importer and processor to build pixel shaders into your game.

Using the Output of the New Processor in Your Game

Try adding a test HLSL source file with a .psh extension to your game project and see how it works.

To test your output

1. Copy into a folder in your game project a simple HLSL source file that you know is free of bugs, and rename the file "Ripple.psh."
2. Right-click on your game project in Solution Explorer, click **Add**, click **Existing Item**, and then click **Ripple.psh**.
3. Once you add the file, right-click it in Solution Explorer, and click **Properties**.

You should now see entries in its **Properties** dialog box assigning **PSImporter** as its content importer and **PSPProcessor** as its content processor. Next time you build your game, Ripple.psh will be built into TestShader.xnb in a form appropriate for your target platform.

4. To use the resulting pixel shader in your game, load it using [ContentManager.Load](#) as follows:

```
PixelShader shader = content.Load<PixelShader>( "TestShader" );
```

Tips for Developing Custom Importers

The following information should help you when you develop content pipeline extensions.

Importing Basic Graphics Objects

The following information should help you import basic graphics objects.

- Make your coordinate system right-handed.

From the standpoint of the observer, the positive x-axis points to the right, the positive y-axis points up, and the positive z-axis points toward you (out from the screen).

- Create triangles that have a clockwise winding order.

The default culling mode removes triangles that have a counterclockwise winding order.

- Call [SwapWindingOrder](#) to change the winding order of a triangle.
- Set the scale for graphical objects to 1 unit = 1 meter.
- Call [TransformScene](#) to change the scale of an object.

Taking Advantage of Content Pipeline Mesh Classes

There are several properties and classes that are particularly useful when using [NodeContent](#) objects to represent a 3D scene or mesh.

- The [NodeContent.Children](#) property represents hierarchical information.
- The [NodeContent.Transform](#) property contains the local transform of the 3d object.
- The [Pipeline.Graphics.MeshContent](#) class (a subclass of [Pipeline.Graphics.NodeContent](#)) is used to represent meshes.

The content pipeline provides two classes that make it easier to create and work with [Pipeline.Graphics.MeshContent](#) objects.

- The [Pipeline.Graphics.MeshBuilder](#) class creates new [Pipeline.Graphics.MeshContent](#) objects, when necessary.
- The [Pipeline.Graphics.MeshHelper](#) class implements useful operations on existing [Pipeline.Graphics.MeshContent](#) objects.

Debugging Custom Importers and Processors

In a way that is similar to projects that create a DLL, content pipeline extension projects cannot be directly run or debugged. However, after completing a few simple steps, you can debug any custom importers and processors used by your game. The following procedure details these steps.

Note

The Start External program: control (located on the Debug page of a project's property pages) is not available in the Microsoft Visual C# Express Edition development environment.

To Debug a Custom Importer or Processor

1. Load an existing XNA Game Studio content pipeline extension project (later referred to as ProjCP) containing the custom importers and/or processors to be debugged.
2. Create a separate test game project (later referred to as "ProjG").
3. In the **References** node of ProjG's nested content project, add a project-to-project reference to ProjCP.
4. Add one or two appropriate items of test content to ProjG, and ensure they are set to use the importer or processor (in ProjCP) you wish to debug.
5. Open the property pages for ProjCP.
6. Click the **Debug** tab, and select **Start external program:**.
7. Enter the path to the local version of MSBuild.exe.
For example, C:\WINDOWS\Microsoft.NET\Framework\v3.5\msbuild.exe.
8. For the **Command line arguments** control, enter the path to ProjG's nested content project.
If this path contains spaces, quote the entire path.
9. Set any required breakpoints in the importer or processor code in ProjCP.
10. Build and debug ProjCP.

Debugging ProjCP causes MSBuild to compile your test content while running under the debugger. This enables you to hit your breakpoints in ProjCP and step through your code.

See Also [Overview of the Content Pipeline](#)
[Content Pipeline Architecture](#)
[Extending an XNA Framework Standard Processor](#)

How To: Extend the Font Description Processor to Support Additional Characters

Describes the process of developing a custom content processor needed to add additional characters to a [FontDescription](#) object based on the text that is required by the game.

In a font description (.spritefont) file, the `<CharacterRegions>` area can be used to add additional characters to a font description. This enables you to use a [SpriteFont](#) to render an additional range of characters.

For some languages, this approach is not ideal. For example, Chinese and Japanese both have many thousands of characters. Adding the full range of characters to `<CharacterRegions>` dramatically increases the size of the font asset and the time required to build the font asset. A better solution adds individual characters whenever the specific characters are needed. You can create a custom content processor to implement this solution.

In this example, a file called *messages.txt* contains all the text rendered by the game. The custom processor adds all the characters contained in the text in this file to a [FontDescription](#). Then it processes the object in the standard way using the base [FontDescriptionProcessor](#) functionality. All the characters in *messages.txt* will then be available to the [SpriteFont](#) object at run time.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download CPFontProcessor_Sample.zip.](#)

Using the Font Description Processor

To specify the character regions and messages to process

1. To add a new Sprite Font called DefaultFont to a game project, go to Solution Explorer, right-click the nested Content node, click **Add**, and then click **New Item**.
2. To add the new sprite font to the game, select the **Sprite Font** template, and then click **Add**.
3. Modify this file to use an existing font and any additional characteristics you prefer.

For more information, see [Sprite Font XML Schema Reference](#).

4. Add a file named *messages.txt* to the game project.
5. Right-click on the game project node in Solution Explorer, click **Add**, and then click **New Item**.
6. Select the **Text File** template, enter **messages.txt** for the file name, and then click **Add** to add the text file to the game.
7. In the new text file, enter any messages that will be printed by the font described in the Sprite Font file.

⚠Caution

We will use the method [File.ReadAllText](#) to read the text in this file. This method requires a carriage return ("`\r`") or line feed ("`\n`") after the last string, so be sure to follow the last line of text in the file with a carriage return or line feed.

To create the new content processor project

The Content Pipeline is part of the build process, and it is separate from your game code. Therefore, you need to create a new assembly that contains the code developed in this topic. Creating this new assembly project is the first step in developing a new processor.

📌Note

It is assumed that you have an existing game project that you will modify. For the purposes of this example, the game project is called "FontGame."

1. To add the new processor project to the game solution, go to Solution Explorer, right-click the **Solution** node, click **Add**, and then click **New Project**.
2. In the dialog box, select the Content Pipeline Extension Library (3.1) template, enter **FontProcessor** in the **Name** field, and then click **OK**. The new project automatically contains references to the XNA Framework run-time and design-time

Content Pipeline assemblies.

To extend the font processor

1. Add the following lines of code, after the last `using` statement:

C#

```
using System.IO;
using System.ComponentModel;
```

2. In `ContentProcessor1.cs`, remove the code near the top of the file, and replace it with the processor and input types.
3. Using attributes, add a processor parameter to the beginning of the class declaration.

This parameter stores the name of the text file that stores the messages displayed by the game.

C#

```
[DefaultValue("messages.txt")]
[DisplayName("Message File")]
[Description("The characters in this file will be automatically added to the font.")]
public string MessageFile
{
    get { return messageFile; }
    set { messageFile = value; }
}
private string messageFile = "..\\messages.txt";
```

4. Change the derivation of `ContentProcessor1` from `ContentProcessor` to `FontDescriptionProcessor`.
5. Modify the `Process` method override to match the following code:

C#

```
public override SpriteFontContent Process(FontDescription input, ContentProcessorContext context)
```

This modification replaces the template parameter and return types with the proper types needed for the extended font processor.

6. Register a Content Pipeline dependency on `messages.txt`.

This dependency tells the Content Pipeline that if `messages.txt` changes, the font must be rebuilt.

C#

```
string fullPath = Path.GetFullPath(MessageFile);

context.AddDependency(fullPath);
```

7. Read the contents of the file, and add each letter to the input font one by one. Note that the `Characters` collection keeps track of duplicates automatically. It is not necessary for the user to make sure that each letter is added only once. The **Characters** collection will contain only one instance of each character, no matter how many times **Add** has been called.

C#

```
string letters = File.ReadAllText(fullPath, System.Text.Encoding.UTF8);

foreach (char c in letters)
{
    input.Characters.Add(c);
}
```

In this example, messages.txt has been saved with Unicode UTF-8 encoding, which is why this encoding format is specified in the call to [File.ReadAllText](#). The default file encoding format for text files that have been added to a Visual Studio project is Western European (Windows) encoding, corresponding to code page 1252. If your text file uses the default encoding, specify the character encoding as follows:

```
string letters = File.ReadAllText( fullPath, System.Text.Encoding.GetEncoding( 1252 ) );
```

8. Call the existing **Process** method of the base [FontDescriptionProcessor](#) to build the font with the newly requested characters.

C#

```
return base.Process(input, context);
```

To associate the custom font processor with the sprite font

1. Compile the solution to build **MyFontProcessor**.

Now you need to add your custom font processor as an available content processor for the game.

2. From **Solution Explorer**, right-click the **Content** node, and then click **Add Reference**.
3. From the **Projects** tab, select your content extension project (**FontProcessor**) node, and click **OK**.

To ensure that the processor project is always up to date when the main game is built, you need to create a project dependency.

4. In Solution Explorer, right-click the game project (**FontGame**) node, and then click **Project Dependencies**.
5. Select the check box next to **FontProcessor**, and then click **OK** to add a new dependency so that **FontGame** depends on **FontProcessor**.
6. Change the content processor for the .spritefont file from **Sprite Font Description - XNA Framework** to the newly created processor.
7. Select the .spritefont file, and then in the **Properties** window, choose your custom processor from the drop-down list associated with the **ContentProcessor** field.

When you build the solution, the new processor adds the characters in the messages.txt file to the list of characters available to the [SpriteFont](#).

Tip

To debug a Content Pipeline importer or processor, add the following line to the processor code to launch the debugger.

```
System.Diagnostics.Debugger.Launch();
```

See Also [XNA Game Studio 3.1](#)

[Extending an XNA Framework Standard Processor](#)

Application Model

Provides functionality to accomplish common game development tasks.

In This Section

[Application Model Overview](#)

The XNA Framework [Game](#) class provides a framework for processing game simulation based on a fixed or variable time interval.

[How To: Load Content](#)

Demonstrates how you can load content and ensure that the content will be reloaded at the appropriate times.

[How To: Allow the Player to Resize a Game Window](#)

Demonstrates how to let the player resize the game window.

[How To: Pause a Game](#)

Demonstrates how to add pause functionality to a game.

[How To: Exit a Game](#)

Demonstrates how to exit a game without finishing the current update.

[How To: Display a Game in Full-Screen Mode](#)

Demonstrates how to start a game in full-screen mode.

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)

Demonstrates how to create a custom [GraphicsDeviceManager](#) that only selects graphics devices with widescreen aspect ratios in full-screen mode.

[How To: Make a Game Time Out](#)

Demonstrates how to make a game time out after a period of inactivity.

[How To: Make a Game Use a Variable Time Step](#)

Demonstrates how to make a game use a variable time step.

Application Model Overview

The XNA Framework [Game](#) class provides a framework for processing game simulation based on a fixed or variable time interval.

This overview covers the following topics.

- [Making a New Game](#)
- [Game Loop Timing](#)
- [Starting the Game](#)
- [Game Components](#)
- [Game Services](#)
- [Game Components Consuming Game Services](#)
- [Tasks](#)

Making a New Game

The first step in creating a new game is to make a class that derives from [Game](#). The new class needs to override [Update](#), [Draw](#), and [Initialize](#). The [Update](#) method is responsible for handling game logic, and the [Draw](#) method is responsible for drawing each frame. The [Initialize](#) method is responsible for game setup before the first frame of the game.

Game Loop Timing

A [Game](#) is either fixed step or variable step, defaulting to fixed step. The type of step determines how often [Update](#) will be called and affects how you need to represent time-based procedures such as movement and animation.

Fixed-Step Game Loops

A fixed-step [Game](#) tries to call its [Update](#) method on the fixed interval specified in [TargetElapsedTime](#). Setting [Game.IsFixedTimeStep](#) to **true** causes a [Game](#) to use a fixed-step game loop. A new XNA project uses a fixed-step game loop with a default [TargetElapsedTime](#) of 1/60th of a second.

In a fixed-step game loop, [Game](#) calls [Update](#) once the [TargetElapsedTime](#) has elapsed. After [Update](#) is called, if it is not time to call [Update](#) again, [Game](#) calls [Draw](#). After [Draw](#) is called, if it is not time to call [Update](#) again, [Game](#) idles until it is time to call [Update](#).

If [Update](#) takes too long to process, [Game](#) sets [IsRunningSlowly](#) to **true** and calls [Update](#) again, without calling [Draw](#) in between. When an update runs longer than the [TargetElapsedTime](#), [Game](#) responds by calling [Update](#) extra times and dropping the frames associated with those updates to catch up. This ensures that [Update](#) will have been called the expected number of times when the game loop catches up from a slowdown. You can check the value of [IsRunningSlowly](#) in your [Update](#) if you want to detect dropped frames and shorten your [Update](#) processing to compensate. You can reset the elapsed times by calling [ResetElapsedTime](#).

When your game pauses in the debugger, [Game](#) will not make extra calls to [Update](#) when the game resumes.

Variable-Step Game Loops

A variable-step game calls its [Update](#) and [Draw](#) methods in a continuous loop without regard to the [TargetElapsedTime](#). Setting [Game.IsFixedTimeStep](#) to **false** causes a [Game](#) to use a variable-step game loop.

Animation and Timing

For operations that require precise timing, such as animation, the type of game loop your game uses (fixed-step or variable-step) is important.

Using a fixed step allows game logic to use the [TargetElapsedTime](#) as its basic unit of time and assume that [Update](#) will be called at that interval. Using a variable step requires the game logic and animation code to be based on [ElapsedGameTime](#) to ensure smooth gameplay. Because the [Update](#) method is called immediately after the previous frame is drawn, the time between calls to [Update](#) can vary. Without taking the time between calls into account, the game would seem to speed up and slow down. The time elapsed between calls to the [Update](#) method is available in the [Update](#) method's *gameTime* parameter. You can reset the elapsed times by calling [ResetElapsedTime](#).

When using a variable-step game loop, you should express rates—such as the distance a sprite moves—in game units per millisecond (ms). The amount a sprite moves in any given update can then be calculated as the rate of the sprite times the elapsed time. Using this approach to calculate the distance the sprite moved ensures that the sprite will move consistently if the speed of the game or computer varies.

Starting the Game

Calling the `Game.Run` method starts a game. This method starts a loop that will call `Update` and `Draw` multiple times a second until `Exit` is called.

Game Components

Game components provide a modular way of adding functionality to a game. You create a game component by deriving the new component either from the `GameComponent` class, or, if the component loads and draws graphics content, from the `DrawableGameComponent` class. You then add game logic and rendering code to the game component by overriding `GameComponent.Update`, `DrawableGameComponent.Draw` and `GameComponent.Initialize`. A game component is registered with a game by passing the component to `Game.Components.Add`. A registered component will have its draw, update, and initialize methods called from the `Game.Initialize`, `Game.Update`, and `Game.Draw` methods.

Game Services

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, `Game.Services`. Service providers register with `Game.Services`, and service consumers request services from `Game.Services`. This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with `Game.Services`. A service is registered by calling `Game.Services.AddService` specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface `IMyService`, you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by `Game.Services.GetService` and specifying the desired service. For example, to retrieve `IGraphicsDeviceService`, you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Services.GetService( typeof(IGraphicsDeviceService) );
```

Game Components Consuming Game Services

The `GameComponent` class provides the `Game` property so a `GameComponent` can determine what `Game` it is attached to. With the `Game` property, a `GameComponent` can call `Game.Services.GetService` to find a provider of a particular service. For example, a `GameComponent` would find the `IGraphicsDeviceService` provider by using the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Game.Services.GetService( typeof( IGraphicsDeviceService ) );
```

Tasks

[Your First Game: Microsoft XNA Game Studio in 2D](#)

How To: Load Content

Demonstrates how you can load content and ensure that the content will be reloaded at the appropriate times.

Note

The methods used to load and unload resources have changed in XNA Game Studio 3.0. [LoadGraphicsContent](#) and [UnloadGraphicsContent](#) have become simply [LoadContent](#) and [UnloadContent](#), and the Boolean parameters are no longer necessary. For backward compatibility, this version still includes [LoadGraphicsContent](#) and [UnloadGraphicsContent](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Loading Content

To load content and ensure it will be reloaded when necessary

1. Derive a class from [Game](#).
2. Override the [LoadContent](#) method of [Game](#).
3. In the [LoadContent](#) method, load your content, including resources loaded by the [ContentManager](#).

C#

```
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    // TODO: Load your game content here
    Box = Content.Load<Model>("box");
}
```

4. Override the [UnloadContent](#) method of [Game](#).
5. In the [UnloadContent](#) method, unload resources that are not managed by the [ContentManager](#).

C#

```
protected override void UnloadContent()
{
    // TODO: Unload any non ContentManager content here
}
```

Loading Content from a Game Library

XNA Game Studio allows code and content to be run from Game Library projects that are added as references to Game projects. If you use a Game Library, you can embed binary resources directly in the Game Library and load them from within. This allows you to distribute code that displays textures, models, or fonts (such as a [DrawableGameComponent](#)) in a .DLL without distributing the .xnb files separately. Note that embedded resources are loaded into memory with the .DLL, and cannot be unloaded from main memory.

To add content to a Game Library

1. Build an existing project containing the content you wish to add.
2. In a library project, choose **Add, New Item** and select "Resources File."
3. If the **Resource Designer** is not opened automatically, double-click the .resx file in the **Solution Explorer**.
4. From the **Resource Designer**, choose **Add Resource, Add Existing File**.

5. Navigate to the "bin\x86\Debug\Content" directory of the project that built the content you wish to add.

This assumes it was built as an x86 Debug project.

6. Select the .xnb file for the content you wish to add to the library.

Make sure the dialog box is displaying "All Files."

Once content has been added to the **Resource Designer**, any code running from within the Library can load the content with a special [ContentManager](#).

To load content from a Game Library

1. Define a new [ContentManager](#).
2. Create a new instance of the [ResourceContentManager](#) class and assign it to your [ContentManager](#).

The second parameter to the [ResourceContentManager](#) constructor identifies the resource project that contains your embedded resources.

C#

```
ContentManager content;
public GameComponent1(Game game)
    : base(game)
{
    content = new ResourceContentManager(game.Services,
        Resource1.ResourceManager);
}
```

3. In the [LoadContent](#) method, load your content normally using your [ContentManager](#).

C#

```
Model box;
protected override void LoadContent()
{
    box = content.Load<Model>("box2");
    base.LoadContent();
}
```

See Also

Concepts

[Application Model Overview](#)

[How To: Draw a Sprite](#)

Reference

[Game Class LoadContent UnloadContent Game Members Microsoft.Xna.Framework Namespace](#)

How To: Allow the Player to Resize a Game Window

Demonstrates how to let the player resize the game window.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Adding Window Resizing Functionality

To add player window resizing to a game

1. Derive a class from [Game](#).
2. Set [Game.GameWindow.AllowUserResizing](#) to **true**.
3. Add an event handler for the [ClientSizeChanged](#) event of [Game.Window](#).

C#

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
    ...
    this.Window.AllowUserResizing = true;
    this.Window.ClientSizeChanged += new EventHandler(Window_ClientSizeChanged);
    ...
}
```

4. Implement a method to handle the [ClientSizeChanged](#) event of [Game.Window](#).

C#

```
void Window_ClientSizeChanged(object sender, EventArgs e)
{
    // Make changes to handle the new window size.
}
```

How To: Pause a Game

Demonstrates how to add pause functionality to a game.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Adding Pause Functionality to a Game

Typically, there are two circumstances when you want to pause your game—at the request of the user, and when the [Guide](#) appears on the screen (obscuring the playfield). A user request could take several different forms. For example, the user might launch a menu, or the user might execute a specific keystroke. You have to decide for yourself how users can pause your game.

Checking for the presence of the [Guide](#) is more straightforward. You can just query the [Guide.IsVisible](#) property.

When the user requests a pause, you don't want to unpaused until the user tells you to do so. When you pause for the [Guide](#), you should unpaused as soon as the [Guide](#) is dismissed.

Pausing your game often means more than just halting your simulation. It could also require pausing or muting any sounds that might be playing, halting controller vibrations, sending a network message, and so on. To handle those tasks, you need to declare a **BeginPause** and **EndPause** method. **EndPause** resumes anything that was halted by **BeginPause**.

To add pause functionality to a game

1. Add a variable to track the pause state.
2. Add a variable to track the state of the pause key.
3. Add a variable to track if the pause is due to the [Guide](#) or to user action.

C#

```
private bool paused = false;
private bool pauseKeyDown = false;
private bool pausedForGuide = false;
```

4. Add a **BeginPause** method to initiate a pause, setting the variables appropriately:

C#

```
private void BeginPause(bool UserInitiated)
{
    paused = true;
    pausedForGuide = !UserInitiated;
    //TODO: Pause audio playback
    //TODO: Pause controller vibration
}
```

5. Add an **EndPause** method to resume from a paused state, resetting variables appropriately:

C#

```
private void EndPause()
{
    //TODO: Resume audio
    //TODO: Resume controller vibration
    pausedForGuide = false;
    paused = false;
}
```

6. Add a function to poll the state of the pause key with `Keyboard.GetState` and `KeyboardState.IsKeyDown`.

If the key has changed from down to up, toggle the pause state using **BeginPause** or **EndPause**.

C#

```
private void checkPauseKey(KeyboardState keyboardState,
    GamePadState gamePadState)
{
    bool pauseKeyDownThisFrame = (keyboardState.IsKeyDown(Keys.P) ||
        (gamePadState.Buttons.Y == ButtonState.Pressed));
    // If key was not down before, but is down now, we toggle the
    // pause setting
    if (!pauseKeyDown && pauseKeyDownThisFrame)
    {
        if (!paused)
            BeginPause(true);
        else
            EndPause();
    }
    pauseKeyDown = pauseKeyDownThisFrame;
}
```

7. Add a function to poll the state of the `Guide`.

If the `Guide` is newly visible, call **BeginPause**.

If the `Guide` is not visible, but the game was paused for the guide, call **EndPause**.

C#

```
private void checkPauseGuide()
{
    // Pause if the Guide is up
    if (!paused && Guide.IsVisible)
        BeginPause(false);
    // If we paused for the guide, unpaue if the guide
    // went away
    else if (paused && pausedForGuide && !Guide.IsVisible)
        EndPause();
}
```

8. During `Update`, check to see if the user paused, or if the `Guide` is active.

Add a conditional around any update code so it will be called only if the game is not paused. Be sure to call **base.Update** even if the simulation is paused.

C#

```
// Check to see if the user has paused or unpaused
checkPauseKey(keyboardState, gamePadState);

checkPauseGuide();

// If the user hasn't paused, Update normally
if (!paused)
{
    Simulate(gameTime);
}
base.Update(gameTime);
```

Remarks

★Best Practice

You might choose to implement a menu of options to display when a game is paused. This might include options to resume, purchase, save, or quit the game. In multiplayer games, pausing will usually disable local input without pausing the game for remote players.

Some situations in which you may want to pause the simulation update or input are when:

- `Guide.IsVisible` is **true**, as the guide will block user inputs from reaching the game.
- `Game.IsActive` is **false**, indicating that trial mode may have ended for the game.
- `GamePadState.IsConnected` is **false**, indicating that the game controller is disconnected.

How To: Exit a Game

Demonstrates how to exit a game without finishing the current update.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Exiting a Game Without Finishing the Current Update

To exit the game loop without running any remaining code in the update handler

1. Derive a class from [Game](#).

You need to create a method that checks [KeyboardState.IsKeyDown](#) for the state of the ESC key.

2. If the ESC key has been pressed, call [Game.Exit](#) and return **true**.

C#

```
bool checkExitKey(KeyboardState keyboardState,
    GamePadState gamePadState)
{
    // Check to see whether ESC was pressed on the keyboard
    // or BACK was pressed on the controller.
    if (keyboardState.IsKeyDown(Keys.Escape) ||
        gamePadState.Buttons.Back == ButtonState.Pressed)
    {
        Exit();
        return true;
    }
    return false;
}
```

3. Call the method in [Game.Update](#), and return from [Update](#) if the method returned **true**.

C#

```
// Check to see if the user has exited
if (checkExitKey(keyboardState, gamePadState))
{
    base.Update(gameTime);
    return;
}
```

4. Create a method to handle the [Game.Exiting](#) event.

The [Exiting](#) event will be issued at the end of the tick in which [Game.Exit](#) is called.

C#

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
    ...
    this.Exiting += new EventHandler(Game1_Exiting);
    ...
}

void Game1_Exiting(object sender, EventArgs e)
```



```
{  
    // Add any code that must execute before the game ends.  
}
```

How To: Display a Game in Full-Screen Mode

Demonstrates how to start a game in full-screen mode.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Starting a Game in Full-Screen Mode

To start a game in full-screen mode

1. Derive a class from [Game](#).
2. After creating the [GraphicsDeviceManager](#), set its [PreferredBackBufferWidth](#) and [PreferredBackBufferHeight](#) to the desired screen width and height.
3. Set [IsFullScreen](#) to **true**.

C#

```
public Game1()
{
    this.graphics.PreferredBackBufferWidth = 1280;
    this.graphics.PreferredBackBufferHeight = 720;

    this.graphics.IsFullScreen = true;
}
```

How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode

Demonstrates how to create a custom [GraphicsDeviceManager](#) that only selects graphics devices with widescreen aspect ratios in full-screen mode.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AspectRatio_Sample.zip.](#)

Restricting Graphics Devices

To restrict graphics devices to widescreen aspect ratios in full-screen mode

1. Create a class that derives from [GraphicsDeviceManager](#).

C#

```
public class CustomGraphicsDeviceManager : GraphicsDeviceManager
{
    public CustomGraphicsDeviceManager( Game game )
        : base( game )
    {
    }
    ...
}
```

2. Add a **WideScreenOnly** property to the class.

The property will be used to turn the widescreen-only behavior on and off.

C#

```
private bool isWideScreenOnly;
public bool IsWideScreenOnly
{
    get { return isWideScreenOnly; }
    set { isWideScreenOnly = value; }
}
```

3. Determine the minimum desired aspect ratio.

C#

```
static float WideScreenRatio = 1.6f; //1.77777779f;
```

4. Override the [RankDevices](#) method of [GraphicsDeviceManager](#).

Note the call to [base.RankDevices](#). This call ensures that the new version of [RankDevices](#) has an already-ranked list of available devices with which to work.

C#

```
protected override void RankDevices(
    List<GraphicsDeviceInformation> foundDevices )
{
    base.RankDevices( foundDevices );
    ...
}
```

5. Add a check to see if the **WideScreenOnly** property is **true**.

C#

```
if (IsWideScreenOnly)
{
    ...
}
```

6. In the **if** block, loop through all of the found devices, and check whether the [PresentationParameters](#) indicate the device is full-screen.
7. If the device is full-screen, determine the aspect ratio of the device by dividing the [BackBufferWidth](#) by the [BackBufferHeight](#).
8. If the aspect ratio is less than the desired aspect ratio, remove the device from the list of found devices.

C#

```
for (int i = 0; i < foundDevices.Count; )
{
    PresentationParameters pp =
        foundDevices[i].PresentationParameters;
    if (pp.IsFullScreen == true)
    {
        float aspectRatio = (float)(pp.BackBufferWidth) /
            (float)(pp.BackBufferHeight);

        // If the device does not have a widescreen aspect
        // ratio, remove it.
        if (aspectRatio < WideScreenRatio)
        {
            foundDevices.RemoveAt( i );
        }
        else { i++; }
    }
    else i++;
}
```

9. Replace the default [GraphicsDeviceManager](#) with the derived [GraphicsDeviceManager](#).
10. To test the new component, set the **WideScreenOnly** and [IsFullScreen](#) properties to **true**.

C#

```
public Game1()
{
    graphics = new CustomGraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    this.graphics.PreferMultiSampling = false;
    this.graphics.PreferredBackBufferWidth = 1280;
    this.graphics.PreferredBackBufferHeight = 720;

    this.graphics.IsFullScreen = true;
    this.graphics.IsWideScreenOnly = true;
    graphics.ApplyChanges();
}
```

How To: Make a Game Time Out

Demonstrates how to make a game time out after a period of inactivity.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Adding Time-Out Functionality to a Game

To make a game time out

1. Create a class that derives from [Game](#).
2. Determine the desired time-out limit in milliseconds.

C#

```
// Time out limit in ms.
static private int TimeoutLimit = 80000;
```

3. Add a variable for tracking the elapsed time since the most recent user activity.

C#

```
// Amount of time that has passed.
private double timeoutCount = 0;
```

4. When user input is checked, set a flag indicating whether any user activity has taken place.

C#

```
GamePadState blankGamePadState = new GamePadState(
    new GamePadThumbSticks(), new GamePadTriggers(),
    new GamePadButtons(), new GamePadDPad());
bool checkActivity(KeyboardState keyboardState,
    GamePadState gamePadState)
{
    // Check to see if the input states are different from last frame
    GamePadState nonpacketGamePadState = new GamePadState(
        gamePadState.ThumbSticks, gamePadState.Triggers,
        gamePadState.Buttons, gamePadState.DPad);

    bool keyidle = keyboardState.GetPressedKeys().Length == 0;
    bool gameidle = blankGamePadState == nonpacketGamePadState;
    if ( keyidle && gameidle)
    {
        // no activity;
        return false;
    }
    return true;
}
```

5. In [Update](#), if there has not been any user activity, increment the tracking variable by the elapsed time since the last call to [Update](#).
6. If there has been some user activity, set the tracking variable to zero.

C#

```
// Check to see if there has been any activity
```

```
if (checkActivity(keyboardState, gamePadState) == false)
{
    timeoutCount += gameTime.ElapsedGameTime.Milliseconds;
}
else
    timeoutCount = 0;
```

7. Check whether the value of the tracking variable is greater than the time-out limit.
8. If the variable is greater than the limit, perform some time-out logic such as playing an idle animation or, in this case, exit the game.

C#

```
// Timeout if idle long enough
if (timeoutCount > TimeOutLimit)
{
    Exit();
    base.Update(gameTime);
    return;
}
```

How To: Make a Game Use a Variable Time Step

Demonstrates how to make a game use a variable time step.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AppModelDemo_Sample.zip.](#)

Making a Game Use a Variable Time Step

To make a game use a variable time step

1. Create a class that derives from [Game](#).
2. Set `IsFixedTimeStep` to **false**.

This causes `Update` to be called as often as possible instead of being called on a fixed interval.

C#

```
this.IsFixedTimeStep = false;
```

3. Since the amount of time between calls to `Update` will vary, specify any rates used in the game as units per millisecond (ms).

C#

```
// Speed in world units per ms.  
private double speed = 0.02f;
```

4. In `Update`, get the value of `gameTime.ElapsedGameTime.TotalMilliseconds`.

This indicates the amount of time that has passed since the last call to `Update`.

C#

```
// Time elapsed since the last call to update.  
double elapsedTime = gameTime.ElapsedGameTime.TotalMilliseconds;
```

5. Determine the change that occurred since the last update by multiplying any rates being used by the elapsed time.

C#

```
// Multiply speed by elapsed time to get the distance moved.  
double distance = (speed * elapsedTime);
```

Graphics

Describes how the XNA Framework Graphics libraries provide low-level resource loading and rendering capabilities.

In This Section

[The XNA Rendering Pipeline](#)

Provides a high-level view of the graphics rendering pipeline for XNA games.

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Describes the relationships between the display, client, viewport, and back buffer size properties.

[Render Targets](#)

Describes how render targets can be an important part of your XNA Framework game when you render simple or complex scenes. For a description of a render target, see [What Is a Render Target?](#).

[Effect States](#)

Provide a mechanism for effects to control the graphics device state. Setting a state through an effect file has exactly the same effect as setting the corresponding render state or sampler state from your C# code.

[2D Graphics](#)

Discusses the basics of 2D rendering and includes examples of how to display sprites.

[3D Graphics](#)

Provides an overview of the 3D Graphics classes as well as tutorials to demonstrate low-level 3D rendering.

["What Is" Articles](#)

Provides brief overviews of some 3D graphics concepts.

See Also

Concepts

[Getting Started with 2D Games at XNA Creators Club Online](#)

[Getting Started with 3D Games at XNA Creators Club Online](#)

[Shader Content Catalog at XNA Creators Club Online](#)

The XNA Rendering Pipeline

Provides a high-level view of the graphics rendering pipeline for XNA games.

The XNA Framework renders graphics by calling into the DirectX 9 rendering pipeline. The following figure shows the subset of the DirectX 9 rendering pipeline used by the XNA Framework:

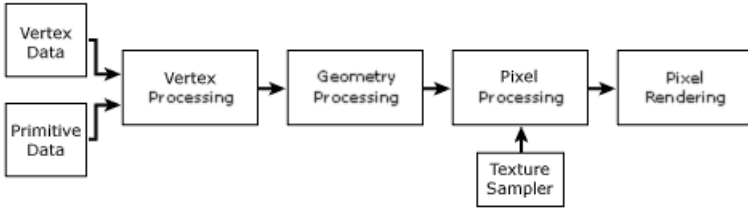


Figure 1. The DirectX 9 rendering pipeline

Pipe line Component	Description	Related Topics
Vertex Data	Vertex memory buffers provide storage for the untransformed model vertices. Typically, you can use a VertexDeclaration to describe what information (position, color, texture coordinates, normals, and so on) is defined for each vertex. Vertex buffers may contain either indexed or non-indexed vertex data.	VertexBuffer , VertexDeclaration
Primitive Data	Geometric primitives, including points, lines, triangles, and polygons, are referenced in the vertex data with index buffers. If a vertex buffer is not indexed, all of the vertices are placed in the vertex buffer in the order they are to be rendered. Because 3D-line lists or triangle lists often reference the same vertices multiple times, this can result in a large amount of redundant data. Index buffers allow you to list each vertex only once in the vertex buffer. An index buffer is a list of indices in the vertex buffer, given in the order that you want the vertices to render.	IndexBuffer How To: Draw Points, Lines, and Other 3D Primitives
Vertex Processing	The vertex shader of an Effect transforms the vertices stored in the vertex buffer. You can use the world, view, and projection matrices defined for the Effect by the game to transform the vertices.	Effect , BasicEffect , HLSL Shader S , How To: Use BasicEffect , How To: Render a Model , How To: Draw a Model with a Custom Effect
Geometry Processing	Clipping, back face culling, attribute evaluation, and rasterization are applied to the transformed vertices. Clipping is the process of removing triangles (or parts of triangles) that do not appear on screen (or into the scissor rectangle if scissor testing is enabled). Back face culling removes triangles that are not facing the camera. Rasterization is the process of assigning pixels to each triangle that remains on screen after the clipping and culling is complete.	ScissorTestEnable , Render Targets , CullMode , Rasterization Rules
Texture Sampler	Texture level-of-detail filtering is applied to textures that will be used by the pixel shader component of an Effect . This includes the TextureAddressMode of each texture coordinate, and the TextureFilter , to use in resizing the texture to fit the object on screen.	SamplerState , What Is Texture Mapping?

Pixel Processing	The pixel shader of an Effect uses geometry data to combine input vertex and texture data with lighting equations, which in turn yields the output pixel color values.	Effect , BasicEffect , HLSL Shader S , How To: Use BasicEffect , How To: Render a Model , How To: Draw a Model with a Custom Effect
Pixel Rendering	Final rendering processes modify pixel color values with alpha, depth, or stencil testing, or by applying alpha blending. All resulting pixel values are presented to the output display.	AlphaTestEnable , DepthBufferEnable , StencilEnable , AlphaBlendEnable , What Is Color Blending? , What Is a Depth Buffer? , What Is a Stencil Buffer?

See Also

Concepts

[3D Graphics Overview](#)

Displays, Client Bounds, Viewports, and Back Buffers

Describes the relationships between the display, client, viewport, and back buffer size properties.

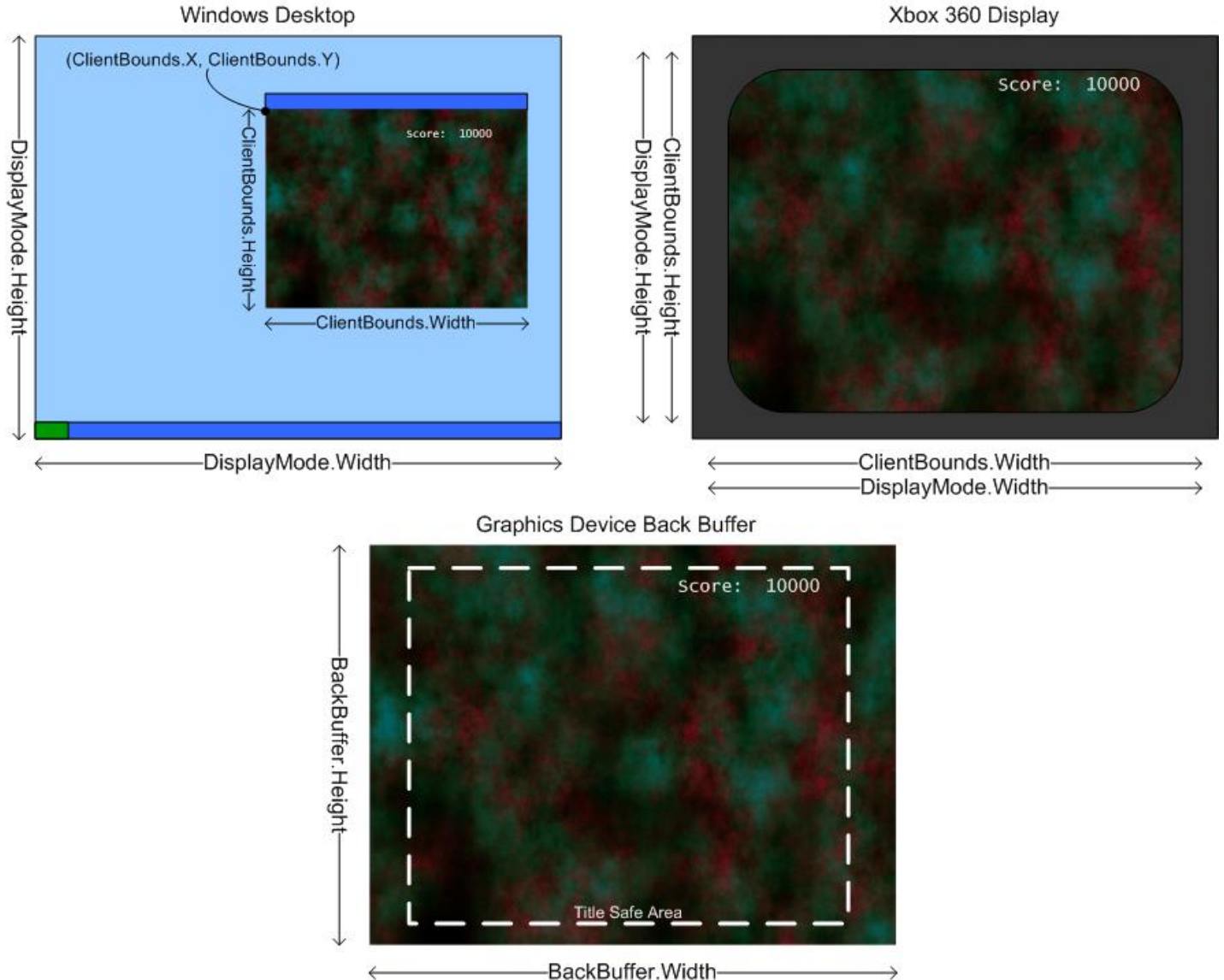


Figure 1. In this diagram, the Windows display is shown with a game that is not in full-screen mode to show the difference between the `DisplayMode` and `ClientBounds` dimensions. Note also that the same back buffer may render differently on an Xbox 360 console compared to a Windows display because the outer 10 to 20 percent of the Xbox 360 back buffer is not "title safe" and may not be visible.

DisplayMode Width and Height

The `DisplayMode Width` and `Height` properties correspond to the *bounds of the display*. The display bounds are different from the bounds of the game window, which can be obtained from the `ClientBounds` dimensions.

When going into full-screen mode on Windows, it is useful to know the display mode width and height so the back buffer dimensions can be set to match the resolution of the display. If the back buffer dimensions are not changed to match the display mode dimensions, the resolution would change when going into full screen mode.

Xbox 360 games should display all text and menu items critical to game play within the inner 80 percent of the x and y resolution for all display modes. This is known as the *title safe region*.

ClientBounds Width and Height

The `ClientBounds Width` and `Height` properties correspond to the *bounds of the game window*, which may be less than the bounds of the display if the game is not full screen. `ClientBounds.X` and `ClientBounds.Y` correspond to the upper-left coordinate of the game window relative to the display mode width and height. On Xbox 360, the `ClientBounds Width` and `Height` properties are always equal to the `DisplayMode Width` and `Height`, and `ClientBounds.X` and `ClientBounds.Y` are always (0, 0).

Viewport Width and Height

The **Viewport Width** and **Height** properties correspond to the current viewport dimensions, which can be thought of as the *view space* width and height. These values are commonly used by methods to create projection matrices, such as [CreatePerspectiveFieldOfView](#). When creating a camera class, you would set the **Viewport** width and height to correspond to the size of the camera viewport.

The dimensions of a viewport default to the dimensions of the render target, but may be a subset of the render target. For example, you might render multiple viewports to a single render target to create a split screen game.

The **Viewport** dimensions do not correspond to the size of the game window (see [ClientBounds](#) for this information) or the display (see [DisplayMode](#) for this information).

Back Buffer Width and Height

The back buffer is a specific render target that has been set to be the next render target presented when [GraphicsDevice.Present](#) is called. To obtain the dimensions of the back buffer, use the [GraphicsDevice.PresentationParameters.BackBufferWidth](#) and [BackBufferHeight](#) properties.

On Windows, the back buffer is created to match the dimensions of the [ClientBounds](#) by default. For Xbox 360 game projects, the back buffer is created with the dimensions that have been specified by the user.

When going into full-screen mode on Windows, it is often desirable to set the back buffer dimensions to match the [DisplayMode](#) dimensions so that the game ("display") resolution does not change when going into full-screen mode.

The graphics back buffer created for an Xbox 360 game project is not necessarily the same size as the final resolution on the television connected to the Xbox 360. The Xbox 360 automatically scales output to the television resolution selected by the user in the System Blade. If the aspect ratio of the back buffer is different than the aspect ratio of the television display mode, the Xbox 360 will automatically add "black bars" (letterboxing) if the user's display is not widescreen.

In addition, if you request a back-buffer resolution that is not supported by the output device, the XNA framework automatically selects the highest resolution supported by the output device. For example, if a graphics back-buffer, with a resolution of 1920x1080 (e.g. 1080p or 1080i), is created and displayed on a device with 480i resolution, the back-buffer is automatically resized to 480i.

See Also

Tasks

[How To: Use Viewports for Split Screen Gaming](#)

Render Targets


Describes how render targets can be an important part of your XNA Framework game when you render simple or complex scenes. For a description of a render target, see [What Is a Render Target?](#)

The following sections detail different aspects of render targets, useful when developing XNA Framework games.

- [Render Targets and Persistence](#)
- [Resolving a Render Target](#)
- [Multiple Render Targets](#)

Render Targets and Persistence

Before XNA Game Studio 3.0, the persistence of render targets depended on the target platform. You can specify the persistence behavior now by using [RenderTargetUsage](#) in the constructor. For example, the following line of code constructs a render target object that is never cleared when set.

 Note
Persistence behavior for the related depth-stencil buffer matches that of the render target.

```
discardTarget= new RenderTarget2D(GraphicsDevice, 200, 200, 1, SurfaceFormat.Color, RenderTargetUsage.PreserveContents);
```

The following tables list related persistence details, based on the target platform and the default value [RenderTargetUsage.DiscardContents](#) used by the constructor.

Creating the Render Target

Platform	Details
Xbox 360	Two surfaces are created: the render target and the resolve texture.
Windows	The default behavior creates two surfaces: the render target surface and the texture used as the output for resolve actions. If you specify persistence (for example, RenderTargetUsage.PreserveContents) and the render target does not have multisampling, you will create only one surface. This surface is a render target texture that is both the surface and the output texture for resolve actions.

Setting the Render Target

Platform	Details
Xbox 360	By default, the resolve texture data is deleted when you set the render target. If you enable persistence, the resolve texture data moves back into the render target (emulating a persisted render target) before you set the new render target.
Windows	The current resolve texture data is saved when you set the render target.

Resolving the Render Target

Platform	Details
Xbox 360	Resolution occurs when you set a new render target. The render target clears to a known, standard color rather than Color.TransparentBlack .
Windows	Resolution occurs when you set a new render target. However, by default, the render target (always a separate surface) is cleared to a known, standard color rather than Color.TransparentBlack . If persistence is enabled, no action is taken.

In addition to [RenderTargetUsage.PreserveContents](#) and [RenderTargetUsage.DiscardContents](#), there is a third option that is platform-specific: [RenderTargetUsage.PlatformContents](#).

This option creates a render target that has persistence for Windows, but not for Xbox 360. However, for games targeting Windows, if you create the render target without multisampling, you only create a single texture. This texture is both the render target surface as well as the resolved texture.

Resolving a Render Target

Before XNA Game Studio 3.0, you called **ResolveRenderTarget** to indicate that you were finished drawing to a render target. This method is now considered obsolete. It causes a compilation error if you use it in your code. The actual resolve operation occurs automatically when you set a new render target (for example, a call to [SetRenderTarget](#)). Note that calling [GetTexture](#) on the current render target throws an exception. This behavior ensures that a new render target has been set (and thus resolved) before you can use the texture for anything else.

Multiple Render Targets

Multiple Render Targets (MRT) refers to the ability to render to multiple surfaces with a single draw call. For games targeting Windows, this option is available on some graphics cards. For games targeting Xbox 360, this option is natively supported. You can use this feature to perform some very nice effects such as deferred shading, or effects such as depth of field or refraction.

The following restrictions apply to MRTs on both platforms.

- All render targets of an MRT must have the same bit depth. However, they can use different formats unless the [SupportsMultipleRenderTargetsIndependentBitDepths](#) property is **false**.
- All surfaces of an MRT should have the same width and height.
- Antialiasing is not supported for MRTs.
- Some MRT implementations do not apply the output write mask. Those that do (indicated by a value of **true** for the [SupportsColorWrite](#) property) have independent color write masks. The number of independent color write masks available is equal to the maximum number of elements the device is capable of supporting.
- Some MRT implementations do not perform post-pixel shader operations, such as dithering, alpha testing, fogging, and blending or masking (except for depth-buffer and depth-stencil testing). Those that do (indicated by a value of **true** for the [SupportsMultipleRenderTargetsPostPixelShaderBlending](#) property).

For those devices that support post-pixel shader operations, use [CheckDeviceFormat](#) (passing [PostPixelShaderBlending](#) for the *queryUsages* parameter value) to verify support for the specific surface format. If the result is **false**, you won't find any available post-pixel shader blending operations for that specific surface format. If **true**, the device is expected to apply the same state to all simultaneous render targets as follows:

- Alpha blend: The color value in αC_i is blended with the i th render target.
- Alpha test: Comparison uses αC_0 . If the comparison fails, the pixel test is terminated for all render targets.
- Fog: Render target 0 is fogged. Other render targets are undefined. Implementations can choose to fog all by using the same state.
- Dithering: Undefined.

See Also
Conceptual
[Graphics](#)

Effect States

Provide a mechanism for effects to control the graphics device state. Setting a state through an effect file has exactly the same effect as setting the corresponding render state or sampler state from your C# code.

The syntax for setting an effect state from an effect file is:

```
effect_state [ [index] ] = expression;
```

Where:

effect_state

A graphics device render state or sampler state to set. A complete list of states is listed below.

[[index]]

Optional integer index. The index identifies a particular state within an array of effect states. The outer brackets indicate that an index is optional. If an index is used, be sure to use inner brackets.

expression

State assignment expression.

⚠ Caution

Effects that set render states that are not defined in this document may compile, but may fail to load due to run-time validation.

For Boolean render or sampler states, nonzero values are converted to **true** (1), and zero values are left alone (**false**).

Effect states can be divided into the following categories.

- [Render States](#)
 - [Pixel Pipe Render States](#)
 - [Vertex Pipe Render States](#)
- [Sampler States](#)
- [Sampler Stage States](#)
- [Shader States](#)
- [Shader Constant States](#)
 - [Pixel Shader Constant States](#)
 - [Vertex Shader Constant States](#)
- [Texture States](#)

Render States

There are two types of render states.

- [Pixel Pipe Render States](#)
- [Vertex Pipe Render States](#)

Pixel Pipe Render States

State	Type	Values
AlphaBlendEnable	bool	true or false . Equivalent to setting AlphaBlendEnable .
AlphaFunc	dword	Any value of CompareFunction . Equivalent to setting AlphaFunction .
AlphaRef	dword	Same values as ReferenceAlpha . Values can range from 0x00000000 through 0x000000FF.
AlphaTestEnable	dword	true or false . Equivalent to setting AlphaTestEnable .
BlendOp	dword	Same values as BlendFunction . Equivalent to setting BlendFunction .
ColorWriteEnable	dword	Bitwise combination of RED GREEN BLUE ALPHA. Equivalent to setting ColorWriteChannels .

DepthBias	int	Same values as DepthBias .
DestBlend	dwor d	Same values as Blend , excluding BothSourceAlpha and BothInverseSourceAlpha , which are not supported.
FillMode	dwor d	Same values as FillMode .
SrcBlend	dwor d	Same values as Blend , excluding BothSourceAlpha and BothInverseSourceAlpha , which are not supported.
StencilEnable	bool	true or false . Equivalent to setting StencilEnable .
StencilFail	dwor d	Keep, Zero, Replace, IncrSat, DecrSat, Invert, Incr, Dec, or TwoSided. Equivalent to setting StencilFail .
StencilFunc	dwor d	Same values as CompareFunction . Equivalent to setting StencilFunction .
StencilMask	dwor d	Same values as StencilMask .
StencilPass	dwor d	Keep, Zero, Replace, IncrSat, DecrSat, Invert, Incr, Dec, or TwoSided. Equivalent to setting StencilPass .
StencilRef	int	Same values as ReferenceStencil .
StencilWriteMask	dwor d	Same values as StencilWriteMask .
StencilZFail	dwor d	Keep, Zero, Replace, IncrSat, DecrSat, Invert, Incr, Dec, or TwoSided. Equivalent to setting StencilDepthBufferFail .
Wrap0–Wrap15	dwor d	Valid values are: <ul style="list-style-type: none"> • COORD0 (which corresponds to TextureWrapCoordinates.Zero) • COORD1 (which corresponds to TextureWrapCoordinates.One) • COORD2 (which corresponds to TextureWrapCoordinates.Two) • COORD3 (which corresponds to TextureWrapCoordinates.Three) • U • V • W
ZEnable	dwor d	true or false .
ZFunc	dwor d	Same values as CompareFunction . Equivalent to setting DepthBufferFunction .
ZWriteEnable	bool	true or false . Equivalent to setting DepthBufferWriteEnable .

Vertex Pipe Render States

State	Type	Values
ClipPlaneEnable	dwor d	<p>Valid values are any DWORD in which the status of each bit (set or not set) toggles the activation state of a corresponding user-defined clipping plane. The least significant bit (bit 0) controls the first clipping plane at index 0, and subsequent bits control the activation of clipping planes at higher indices. If a bit is set, the system applies the appropriate clipping plane during scene rendering. The default value is 0.</p> <p>User-defined clipping planes are enabled when the value set in this render state contains one or more set bits (that is, is not 0). The value of the render state is not important; the system does not interpret the value as a number. Rather, the value enables the clipping plane, whose corresponding bit is set. Bit 0 controls the state of the first clipping plane (at index 0), bit 1 the second plane, and so on.</p> <p>The bit patterns that these macros create can be combined by using a logical OR operation to simultaneously enable multiple clipping planes. Omitting one of these from the combination effectively disables the clipping plane at that index.</p>

Cull Mode	dw	Same values as CullMode .
MultiSampleAntiAlias	bool	true or false . Equivalent to setting MultiSampleAntiAlias .
MultiSampleMask	dw	Same values as MultiSampleMask .
PointSize	float	
PointSize_Min	float	
PointSize_Max	float	
PointSizeEnable	bool	True or False.

Sampler States

State	Type	Values
Sampler	sampler	null , or a sampler state block.

Sampler Stage States

Sampler stages are used to sample textures. Sampler state determines filtering types and texture addressing modes.

Sampler state	Type	Values
AddressU[16]	dw ord	Same values as TextureAddressMode . Equivalent to setting AddressU .
AddressV[16]	dw ord	Same values as TextureAddressMode . Equivalent to setting AddressV .
AddressW[16]	dw ord	Same values as TextureAddressMode . Equivalent to setting AddressW .
BorderColor[16]	DWORD	Either 0x00000000 (black) or 0xFFFFFFFF (white). Xbox 360 only supports a border color of either black or white. For Xbox 360, nonzero values are changed to 0xFFFFFFFF; zero values are left as is.
MagFilter[16]	dw ord	Point, Linear, or Anisotropic. Equivalent to setting MagFilter .
MaxAnisotropy[16]	dw ord	Values must be in the range [1, 16]. The default value is 1.
MaxMipLevel[16]	int	Values range from 0 to $(n - 1)$ where 0 is the largest. The default value is zero.
MinFilter[16]	dw ord	Point, Linear, or Anisotropic. Equivalent to setting MinFilter .
MipFilter[16]	dw ord	None, Point, or Linear. Equivalent to setting MipFilter .

MipMapLodBias[16]	float	Values must be in the range [-16.0f, 15.9375f]. This value is converted into a 5.5 fixed-point format, where 5 bits are used for a signed integer and 5 bits are reserved for the fractional part. The default value is zero.
-------------------	--------------	---

Shader States

There are only two effect shader states: one associated with a vertex shader object and/or associated with a pixel shader object.

Pixel Shader States

State	Type	Values
PixelShader	pixelshader	NULL, an assembly block, a compile target, or a pixel shader parameter.

Vertex Shader States

State	Type	Values
VertexShader	vertexshader	NULL, an assembly block, a compile target, or a pixel shader parameter.

Shader Constant States

Shader constant states are used for setting shader constant registers.

Pixel Shader Constant States

State	Type	Values
PixelShaderConstant	float [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of floats ; <i>m</i> and <i>n</i> are optional.
PixelShaderConstant1	float4	One 4D float .
PixelShaderConstant2	float4x2	Two 4D floats .
PixelShaderConstant3	float4x3	Three 4D floats .
PixelShaderConstant4	float4x4	Four 4D floats .
PixelShaderConstantB	bool [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of bools ; <i>m</i> and <i>n</i> are optional.
PixelShaderConstantI	int [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of ints ; <i>m</i> and <i>n</i> are optional.
PixelShaderConstantF	float [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of floats ; <i>m</i> and <i>n</i> are optional.

Vertex Shader Constant States

These shader states set vertex shader constants.

State	Type	Values
VertexShaderConstant	float [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of floats ; <i>m</i> and <i>n</i> are optional.
VertexShaderConstant1	float4	One 4D float .
VertexShaderConstant2	float4x2	Two 4D floats .
VertexShaderConstant3	float4x3	Three 4D floats .
VertexShaderConstant4	float4x4	Four 4D floats .
VertexShaderConstantB	bool [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of bools ; <i>m</i> and <i>n</i> are optional.
VertexShaderConstantI	int [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of ints ; <i>m</i> and <i>n</i> are optional.
VertexShaderConstantF	float [<i>m</i>][<i>n</i>]	<i>m</i> × <i>n</i> array of floats ; <i>m</i> and <i>n</i> are optional.

Texture States

State	Type	Values
Texture[8]	texture	NULL, or a texture parameter.

See Also

Concepts

[Shader Content Catalog at XNA Creators Club Online](#)

2D Graphics

Discusses the basics of 2D rendering and includes examples of how to display sprites.

In This Section

[2D Graphics Overview](#)

Summarizes the basics of using sprites.

[How To: Draw a Sprite](#)

Demonstrates how to draw a sprite by using the [SpriteBatch](#) class.

[How To: Animate a Sprite](#)

Demonstrates how to animate a sprite from a texture using a custom class.

[How To: Draw a Masked Sprite over a Background](#)

Demonstrates how to draw a foreground and background sprite using the [SpriteBatch](#) class, where only part of the foreground sprite masks the background.

[How To: Make a Scrolling Background](#)

Demonstrates how to draw a scrolling background sprite using the [SpriteBatch](#) class.

[How To: Rotate a Sprite](#)

Demonstrates how to rotate a sprite around its center.

[How To: Rotate a Group of Sprites](#)

Demonstrates how to rotate a group of sprites around a single point using a rotation [Matrix](#).

[How To: Scale a Sprite](#)

Demonstrates how to scale a sprite using a uniform scale, nonuniform scale, or a destination rectangle.

[How To: Tile a Sprite](#)

Demonstrates how to draw a sprite repeatedly in the x and y directions in one [Draw](#) call.

[How To: Tint a Sprite](#)

Demonstrates how to tint a sprite using a [Color](#) value.

[How To: Scale Sprites Based On Screen Size](#)

Demonstrates how to scale sprites using a matrix that is created based on the viewport width.

[How To: Draw Point Sprites](#)

Demonstrates how to create and draw point sprites.

[How To: Apply a Pixel Shader to Sprites](#)

Demonstrates how to apply a pixel shader to sprites.

[How To: Draw a Sprite Over a Model](#)

Demonstrates how to draw a sprite so that it obscures a model.

[How To: Draw Text](#)

Demonstrates how to import a [SpriteFont](#) into a project and draw text using [DrawString](#).

Concepts

[Getting Started with 2D Games at XNA Creators Club Online](#)

2D Graphics Overview

Sprites are 2D bitmaps drawn directly on the screen, as opposed to being drawn in 3D space. Sprites are commonly used to display information such as health bars, number of lives, or text such as scores. Some games, especially older games, are composed entirely of sprites. This topic discusses sprites in detail, covering the following areas.

- [Overview](#)
- [Sprite Origin](#)
- [Sprite Depth](#)
- [Sampling Textures](#)
- [Sprite Scaling](#)
- [Sprite Transformation Matrices](#)
- [Sprite Fonts](#)
- [Sprite Batching](#)
- [Render States](#)
- [Point Sprites](#)

Overview

Sprites are positioned on the screen by coordinates. The width and height of the screen is the same as the back buffer. The x-axis represents the screen width and the y-axis represents the screen height. It is important to note that the y-axis is measured from the top of the screen and increases as you move *down* the screen, and the x-axis is measured from left to right. For example, when the graphics back buffer is 800×600, 0,0 is the upper left of the screen, and 800,600 is the lower right of the screen (see Figure 1).

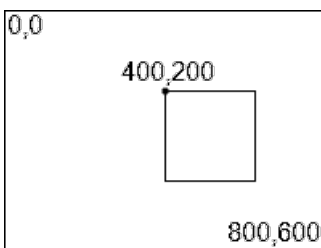


Figure 1. A sprite's location in screen coordinates (x-axis 400, y-axis 200)

To draw a sprite, you must create a [SpriteBatch](#) object, initialize it by calling [Begin](#), and then call [Draw](#) for each sprite. The bitmap information for a sprite is taken from a [Texture2D](#) object. The texture may contain alpha channel information to make part of the texture transparent or semi-transparent. You can tint, rotate, or scale sprites by using [Draw](#). This method also gives you the option of drawing only part of the texture on the screen. After you draw the sprites, call [End](#) before calling [Present](#).

Sprite Origin

When you draw sprites, the sprite *origin* is the most important concept. The origin is a specific point on the sprite, which is by default the upper-left corner of the sprite, or (0,0). [Draw](#) draws the origin of the sprite at the screen location you specify. For example, if you draw a 50×50 pixel sprite at location (400,200) without specifying an origin, the upper left of the sprite will be on pixel (400,200). If you use the center of the 50×50 sprite as the origin (25,25), to draw the sprite in the same position you must add the origin coordinates to the position. In this case, the position is (425,225) and the origin is (25,25), as shown in Figure 2.

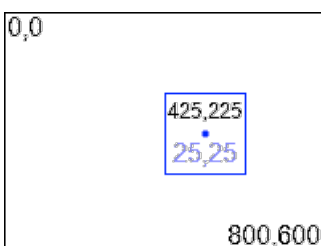


Figure 2. The blue dot indicates the center coordinate of the sprite

When rotating a sprite, the method uses the origin as the center of the rotation. In these cases, it is common to use the center of the sprite as the origin when calculating where to draw the sprite on the screen.

Sprite Depth

Sprites also have a concept of *depth*. When drawing a sprite, you can specify a depth between 0 and 1 as a floating-point number. Sprites drawn at a depth of 0 are at the "front" of the screen, and will cover any sprites drawn at a lower depth. Sprites drawn at a depth of 1 are at the "back" of the screen, and will be covered by any sprites drawn at a depth less than 1.

Sampling Textures

A sprite is based on a [Texture2D](#) object—in other words, a bitmap. [Draw](#) can draw the entire texture, or a portion of the texture. To draw a portion of the texture, use the **sourceRectangle** parameter to specify which *texels* to draw as a sprite. A *texel* is a pixel in the texture. A 32×32 texture has 1024 texels, specified as x and y values similar to how screen coordinates are specified. Specifying a **sourceRectangle** of (0, 0, 16, 16) would select the upper-left quadrant of a 32×32 texture.

Sprite Scaling

[Draw](#) provides three methods of scaling sprites. [Draw](#) accepts either a uniform scaling parameter, a nonuniform scaling parameter, or a source and destination rectangle. The uniform scaling parameter is a floating-point number that multiplies the sprite size through both the x- and y-axes. This will shrink or expand the sprite along each axis equally, maintaining the original ratio between the sprite width and height.

To scale the x- and y-axes independently, [Draw](#) accepts a [Vector2](#) value as a scalar. This [Vector2](#) specifies nonuniform scaling: x- and y-axes are scaled independently according to the **X** and **Y** fields of the [Vector2](#).

[Draw](#) also accepts a source and destination rectangle. The destination rectangle is specified in screen coordinates, while the source rectangle is specified in texels. [Draw](#) takes the pixels on the texture specified in **sourceRectangle** and scales them independently along the x- and y-axes until they fit the screen coordinates specified by **destinationRectangle**.

Sprite Transformation Matrices

XNA Game Studio includes a feature for sprite batches—the ability to specify a transformation matrix that the batch can apply to each sprite before drawing. The transformation matrix can be any combination of translation, rotation, or scaling matrices multiplied together into a single matrix. This matrix is combined with the sprite position, rotation, scaling, and depth parameters supplied to [Draw](#). Because the matrix also applies to depth, any z-coordinate transformation that makes the sprite depth greater than 1.0 or less than 0.0 will cause the sprite to disappear.

See [How To: Rotate a Group of Sprites](#) for an example of matrix rotation, and [How To: Scale Sprites Based On Screen Size](#) for an example of matrix scaling.

Sprite Fonts

XNA Game Studio enables you to draw text using [SpriteBatch](#). The [DrawString](#) method will draw text on screen with position, color, rotation, origin, and scaling. [DrawString](#) also requires a special type of texture encapsulated by the [SpriteFont](#) class. A [SpriteFont](#) is created by the Content Pipeline when you add a Sprite Font file to your project. The Sprite Font file has information such as the name and point size of the font, and which Unicode characters to include in the [SpriteFont](#) texture. At run time, a [SpriteFont](#) is loaded with [ContentManager.Load](#) just like a [Texture2D](#) object.

See [How To: Draw Text](#) for an example of how to use [SpriteFont](#) to draw text, and [Sprite Font XML Schema Reference](#) for a list of Sprite Font tags. You can use the Content Pipeline to determine your character regions automatically. For more information, see [How To: Extend the Font Description Processor to Support Additional Characters](#).

Sprite Batching

In normal drawing, the [SpriteBatch](#) object does not change any render states or draw any sprites until you call [End](#). This is known as *Deferred* mode. In Deferred mode, [SpriteBatch](#) saves the information from each [Draw](#) call until you call [End](#). When you then call [End](#), [SpriteBatch](#) changes the graphics device settings and draws each sprite in the batch. [End](#) then resets the device settings, if you specified [SaveStateMode.SaveState](#).

If you call [Begin](#), specifying [SpriteSortMode.Immediate](#), it triggers *Immediate* mode. In Immediate mode, the [SpriteBatch](#) immediately changes the graphics device render states to begin drawing sprites. Thereafter, each call to [Draw](#) immediately draws the sprite using the current device settings. Calling [End](#) resets the device settings, if you specified [SaveStateMode.SaveState](#).

In Immediate mode, once you call [Begin](#) on one [SpriteBatch](#) instance, do not call it on any other [SpriteBatch](#) instance until you call [End](#) for the first [SpriteBatch](#).

Deferred mode is slower than Immediate mode, but it allows multiple instances of [SpriteBatch](#) to accept [Begin](#) and [Draw](#) calls without interfering with each other.

Render States

The [SpriteBatch](#) object sets the following render states on the graphics card when drawing sprites.

Render State	Value
GraphicsDevice.RenderState.CullMode	CullMode.CullCounterClockwiseFace
GraphicsDevice.RenderState.DepthBufferEnable	false
GraphicsDevice.RenderState.AlphaBlendEnable	true
GraphicsDevice.RenderState.AlphaBlendOperation	BlendFunction.Add
GraphicsDevice.RenderState.SourceBlend	Blend.SourceAlpha
GraphicsDevice.RenderState.DestinationBlend	Blend.InverseSourceAlpha
GraphicsDevice.RenderState.SeparateAlphaBlendEnabled	false
GraphicsDevice.RenderState.AlphaTestEnable	true
GraphicsDevice.RenderState.AlphaFunction	CompareFunction.Greater
GraphicsDevice.RenderState.ReferenceAlpha	0
GraphicsDevice.SamplerStates[0].AddressU	TextureAddressMode.Clamp
GraphicsDevice.SamplerStates[0].AddressV	TextureAddressMode.Clamp
GraphicsDevice.SamplerStates[0].MagFilter	TextureFilter.Linear
GraphicsDevice.SamplerStates[0].MinFilter	TextureFilter.Linear
GraphicsDevice.SamplerStates[0].MipFilter	TextureFilter.Linear
GraphicsDevice.SamplerStates[0].MipMapLevelOfDetailBias	0.0f
GraphicsDevice.SamplerStates[0].MaxMipLevel	0

The most important settings changed are [DepthBufferEnable](#) (normally **true**), [AlphaBlendEnable](#) (normally **false**), and [AlphaTestEnable](#) (normally **false**). You may also want to set [TextureAddressMode.Wrap](#) for the [AddressU](#) and [AddressV](#) sampler states.

The [SpriteBatch](#) object also sets the [Vertices](#), [Indices](#), [VertexDeclaration](#), [VertexShader](#), and [PixelShader](#) properties on the current [GraphicsDevice](#).

In Immediate mode, all of the render states can be changed before [Draw](#) is called, and [Draw](#) will use the new render states. [How To: Apply a Pixel Shader to Sprites](#) takes advantage of this to apply a custom pixel shader to [Draw](#).

Point Sprites

Point sprites are sprites drawn in 3D space. These sprites are specified by a position and a size. When a point sprite is rendered, it always appears the same size on screen (no matter where the point lies in 3D space relative to the camera) and faces the camera. Point sprites are used to render particle systems. [How To: Apply a Pixel Shader to Sprites](#) gives an example of how to render point sprites using a simple pixel and vertex shader.

See Also

Tasks

- [How To: Draw a Sprite](#)
- [How To: Draw Text](#)
- [How To: Animate a Sprite](#)
- [How To: Draw a Masked Sprite over a Background](#)
- [How To: Make a Scrolling Background](#)
- [How To: Rotate a Sprite](#)
- [How To: Rotate a Group of Sprites](#)
- [How To: Scale a Sprite](#)
- [How To: Tint a Sprite](#)
- [How To: Scale Sprites Based On Screen Size](#)
- [How To: Draw Point Sprites](#)
- [How To: Apply a Pixel Shader to Sprites](#)

Concepts

[Getting Started with 2D Games at XNA Creators Club Online](#)

Reference

- [SpriteBatch](#)
- [Draw](#)
- [Texture2D](#)

How To: Draw a Sprite

Demonstrates how to draw a sprite by using the [SpriteBatch](#) class.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download SpriteDemo_Sample.zip.](#)

Drawing a Sprite

To draw a sprite on screen

1. Derive a class from [Game](#).
2. Define a [SpriteBatch](#) object as a field on your game class.
3. In your [LoadContent](#) method, construct the [SpriteBatch](#) object, passing the current graphics device.
4. Load the textures that will be used for drawing sprites in [LoadContent](#).

In this case, the example uses the **Content** member to load a texture from the XNA Framework Content Pipeline. The texture must be in the project, with the same name passed to [Load](#).

C#

```
private Texture2D SpriteTexture;
private Rectangle TitleSafe;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    SpriteTexture = Content.Load<Texture2D>("Sprite");
    TitleSafe = GetTitleSafeArea(.8f);
}
```

5. In the overridden [Draw](#) method, call [Clear](#).
6. After [Clear](#), call [Begin](#) on your [SpriteBatch](#) object.
7. Create a [Vector2](#) to represent the screen position of the sprite.

On Xbox 360, be careful not to draw foreground sprites on the outer 10 or 20 percent of the screen. Some televisions may obscure the edge of the screen. The **GetTitleSafeArea** function calculates the area of the current [Viewport](#) that is safe, given a specified safety percentage.

C#

```
protected Rectangle GetTitleSafeArea( float percent )
{
    Rectangle retval = new Rectangle(
        graphics.GraphicsDevice.Viewport.X,
        graphics.GraphicsDevice.Viewport.Y,
        graphics.GraphicsDevice.Viewport.Width,
        graphics.GraphicsDevice.Viewport.Height );

    #if XBOX
        // Find Title Safe area of Xbox 360.
        float border = (1 - percent) / 2;
        retval.X = (int)(border * retval.Width);
        retval.Y = (int)(border * retval.Height);
        retval.Width = (int)(percent * retval.Width);
        retval.Height = (int)(percent * retval.Height);
        return retval;
    #endif
}
```

```
#else
    return retval;
#endif
}
```

8. Call `Draw` on your `SpriteBatch` object, passing the texture to draw, the screen position, and the color to apply.
9. Use `Color.White` to draw the texture without any color effects.
10. When all the sprites have been drawn, call `End` on your `SpriteBatch` object.

C#

```
protected override void Draw( gameTime )
{
    graphics.GraphicsDevice.Clear( Color.CornflowerBlue );
    spriteBatch.Begin();
    Vector2 pos = new Vector2( TitleSafe.Left, TitleSafe.Top );
    spriteBatch.Draw(SpriteTexture, pos, Color.White);
    spriteBatch.End();
    base.Draw( gameTime );
}
```

See Also

Concepts

[2D Graphics Overview](#)

[Xbox 360 Programming Considerations](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Animate a Sprite

Demonstrates how to animate a sprite from a texture using a custom class.

The source code in this example assumes the texture you are loading is a *strip* of equal-sized images. In this example, the texture is a 256×64 texture with four frames. This example uses a class named **AnimatedTexture**, which is included with the sample.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AnimatedSprite_Sample.zip.](#)

Drawing an Animated Sprite

To draw an animated sprite

1. Follow the steps of [How To: Draw a Sprite](#).
2. In your game's constructor, create an instance of the **AnimatedTexture** class.

This example uses (0,0) as the origin of the texture, no rotation, a scale of 2, and a depth of 0.5.

C#

```
private AnimatedTexture SpriteTexture;
private const float Rotation = 0;
private const float Scale = 2.0f;
private const float Depth = 0.5f;
public Game1()
{
    ...
    SpriteTexture = new AnimatedTexture(Vector2.Zero,
        Rotation, Scale, Depth);
#if ZUNE
    // Frame rate is 30 fps by default for Zune.
    TargetElapsedTime = TimeSpan.FromSeconds(1 / 30.0);
#endif
}
```

3. Load the texture or textures that provide the image data for the animation.

In this example, the **AnimatedTexture** class loads a single texture and divides it into frames of animation. It uses the last parameter to determine how many frames to draw each second. In this case, it draws four frames at 2 fps (frames per second).

C#

```
private Viewport viewport;
private Vector2 shipPos;
private const int Frames = 4;
private const int FramesPerSec = 2;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    // "shipanimated" is the name of the sprite asset in the project.
    SpriteTexture.Load(Content, "shipanimated", Frames, FramesPerSec);
    viewport = graphics.GraphicsDevice.Viewport;
    shipPos = new Vector2(viewport.Width / 2, viewport.Height / 2);
}
```

4. In your game's `Update` method, determine which frame of animation to display.

C#

```
protected override void Update(GameTime gameTime)
{
    ...
    float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;

    // TODO: Add your game logic here.
    SpriteTexture.UpdateFrame(elapsed);
    base.Update(gameTime);
}
```

This is handled by `AnimatedTexture`'s **UpdateFrame** method, which takes the elapsed seconds between updates as a parameter.

C#

```
// class AnimatedTexture
public void UpdateFrame(float elapsed)
{
    if (Paused)
        return;
    TotalElapsed += elapsed;
    if (TotalElapsed > TimePerFrame)
    {
        Frame++;
        // Keep the Frame between 0 and the total frames, minus one.
        Frame = Frame % framecount;
        TotalElapsed -= TimePerFrame;
    }
}
```

5. In your game's `Draw` method, call `Draw` on the **AnimatedTexture** object.

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    // TODO: Add your drawing code here
    spriteBatch.Begin();
    SpriteTexture.DrawFrame(spriteBatch, shipPos);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

AnimatedTexture will draw the sprite using the subrectangle of the texture that contains the desired animation.

C#

```
// class AnimatedTexture
public void DrawFrame(SpriteBatch batch, Vector2 screenPos)
{
    DrawFrame(batch, Frame, screenPos);
}
public void DrawFrame(SpriteBatch batch, int frame, Vector2 screenPos)
```

```
{  
    int FrameWidth = myTexture.Width / framecount;  
    Rectangle sourcerect = new Rectangle(FrameWidth * frame, 0,  
        FrameWidth, myTexture.Height);  
    batch.Draw(myTexture, screenPos, sourcerect, Color.White,  
        Rotation, Origin, Scale, SpriteEffects.None, Depth);  
}
```

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Draw a Masked Sprite over a Background

Demonstrates how to draw a foreground and background sprite using the [SpriteBatch](#) class, where only part of the foreground sprite masks the background.

The foreground sprite in this example must include masking information.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download BackgroundSprite_Sample.zip.](#)

Drawing a Foreground and Background Sprite

To draw a foreground and background sprite

1. Create the game class, and load your content as described in the procedures of [How To: Draw a Sprite](#).

C#

```
private Vector2 ViperPos; // Position of foreground sprite on screen
private int ScrollHeight; // Height of background sprite
private Viewport viewport;
Texture2D ShipTexture;
Texture2D StarTexture;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    StarTexture = Content.Load<Texture2D>("starfield");
    ShipTexture = Content.Load<Texture2D>("ship");
    viewport = graphics.GraphicsDevice.Viewport;

    ViperPos.X = viewport.Width / 2;
    ViperPos.Y = viewport.Height - 100;
    ScrollHeight = StarTexture.Height;
}
```

2. In [Draw](#), call [Begin](#) for the [SpriteBatch](#).
3. Specify [SpriteBlendMode.None](#).

This will tell the [SpriteBatch](#) to ignore alpha color values when drawing sprites. By default, the z-order of sprites is the order in which they are drawn.

4. Draw the background sprites, and then call [End](#).

C#

```
spriteBatch.Begin(SpriteBlendMode.None);
DrawBackground(spriteBatch);
spriteBatch.End();
```

5. Call [Begin](#) for the [SpriteBatch](#) again.
6. This time, specify [SpriteBlendMode.AlphaBlend](#).

This will cause pixels on the sprite with an alpha value less than 255 to become progressively transparent based on the magnitude of the alpha value. An alpha of 0 will make the pixel completely transparent. Calling [Begin](#) with no parameters causes [SpriteBatch](#) to default to [SpriteBlendMode.AlphaBlend](#).

7. Draw the foreground sprites, then call [End](#).

C#

```
spriteBatch.Begin(SpriteBlendMode.AlphaBlend);  
DrawForeground(spriteBatch);  
spriteBatch.End();
```

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

[What Is Color Blending?](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Make a Scrolling Background

Demonstrates how to draw a scrolling background sprite using the [SpriteBatch](#) class.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ScrollingBackground_Sample.zip.](#)

Drawing a Scrolling Background Sprite

To draw a scrolling background sprite

1. Create the game class.
2. Load resources as described in the procedures of [How To: Draw a Sprite](#).
3. Load the background texture.

C#

```
private ScrollingBackground myBackground;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    myBackground = new ScrollingBackground();
    Texture2D background = Content.Load<Texture2D>("starfield");
    myBackground.Load(GraphicsDevice, background);
}
```

4. Determine the size of the background texture and the size of the screen.

The texture size is determined using the [Height](#) and [Width](#) properties, and the screen size is determined using the [Viewport](#) property on the graphics device.

5. Using the texture and screen information, set the origin of the texture to the center of the top edge of the texture, and the initial screen position to the center of the screen.

C#

```
// class ScrollingBackground
private Vector2 screenpos, origin, texturesize;
private Texture2D mytexture;
private int screenheight;
public void Load( GraphicsDevice device, Texture2D backgroundTexture )
{
    mytexture = backgroundTexture;
    screenheight = device.Viewport.Height;
    int screenwidth = device.Viewport.Width;
    // Set the origin so that we're drawing from the
    // center of the top edge.
    origin = new Vector2( mytexture.Width / 2, 0 );
    // Set the screen position to the center of the screen.
    screenpos = new Vector2( screenwidth / 2, screenheight / 2 );
    // Offset to draw the second texture, when necessary.
    texturesize = new Vector2( 0, mytexture.Height );
}
```

6. To scroll the background, change the screen position of the background texture in your [Update](#) method.

This example moves the background down 100 pixels per second by increasing the screen position's Y value.

C#

```
protected override void Update(GameTime gameTime)
{
    ...
    // The time since Update was called last.
    float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;

    // TODO: Add your game logic here.
    myBackground.Update(elapsed * 100);

    base.Update(gameTime);
}
```

The Y value is kept no larger than the texture height, making the background scroll from the bottom of the screen back to the top.

C#

```
public void Update( float deltaY )
{
    screenpos.Y += deltaY;
    screenpos.Y = screenpos.Y % mytexture.Height;
}
// ScrollingBackground.Draw
```

7. Draw the background using the origin and screen position calculated in [LoadContent](#) and [Update](#).

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();
    myBackground.Draw(spriteBatch);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

In case the texture doesn't cover the screen, another texture is drawn. This subtracts the texture height from the screen position using the **texture.size** vector created at load time. This creates the illusion of a loop.

C#

```
public void Draw( SpriteBatch batch )
{
    // Draw the texture, if it is still onscreen.
    if (screenpos.Y < screenheight)
    {
        batch.Draw( mytexture, screenpos, null,
            Color.White, 0, origin, 1, SpriteEffects.None, 0f );
    }
    // Draw the texture a second time, behind the first,
    // to create the scrolling illusion.
    batch.Draw( mytexture, screenpos - texture.size, null,
        Color.White, 0, origin, 1, SpriteEffects.None, 0f );
}
```

See Also

Tasks

[How To: Draw a Sprite](#)

[How To: Draw a Masked Sprite over a Background](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Rotate a Sprite

Demonstrates how to rotate a sprite around its center. The source code in this example assumes that the texture being loaded is 64×64 pixels.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download RotateSprite_Sample.zip.](#)

Drawing a Rotated Sprite

To draw a rotated sprite on screen

1. Follow the procedures of [How To: Draw a Sprite](#).
2. Determine the screen location of the sprite, and the point within the texture that will serve as the origin.

By default, the origin of a texture is (0,0), the upper-left corner. When you draw a sprite, the origin point in the texture is placed on the screen coordinate specified by the *at* parameter. In this example, the origin is the center of the texture, and the screen position is the center of the screen.

C#

```
private Texture2D SpriteTexture;
private Vector2 origin;
private Vector2 screenpos;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    SpriteTexture = Content.Load<Texture2D>("ship");
    Viewport viewport = graphics.GraphicsDevice.Viewport;
    origin.X = SpriteTexture.Width / 2;
    origin.Y = SpriteTexture.Height / 2;
    screenpos.X = viewport.Width / 2;
    screenpos.Y = viewport.Height / 2;
}
```

3. In your [Update](#) method, determine the rotation angle to use for the sprite.

The angle is specified in radians, and it can be greater than two times π , but does not need to be.

C#

```
private float RotationAngle;
protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
        this.Exit();

    // The time since Update was called last.
    float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;

    // TODO: Add your game logic here.
    RotationAngle += elapsed;
    float circle = MathHelper.Pi * 2;
    RotationAngle = RotationAngle % circle;

    base.Update(gameTime);
}
```

```
}
```

4. In your own [Draw](#) method, call [SpriteBatch.Draw](#) with the texture, angle, screen position, and origin of the texture.

C#

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    // TODO: Add your drawing code here
    spriteBatch.Begin();
    spriteBatch.Draw(SpriteTexture, screenpos, null, Color.White, RotationAngle,
        origin, 1.0f, SpriteEffects.None, 0f);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

5. When all the sprites have been drawn, call [End](#) on your [SpriteBatch](#) object.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Rotate a Group of Sprites

Demonstrates how to rotate a group of sprites around a single point using a rotation [Matrix](#). It uses two methods. It rotates the sprite positions manually during [Update](#), and it creates a rotation matrix during [Update](#) that [SpriteBatch](#) can use during [Draw](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download RotateSpriteGroup_Sample.zip.](#)

Drawing a Rotated Group of Sprites

To draw a rotated group of sprites on screen

1. Follow the steps of [How To: Draw a Sprite](#).
2. Create one set of [Vector2](#) objects that represents the unrotated positions of the sprites, and one set to hold the rotated values.

C#

```
private Vector2[] myVectors;
private Vector2[] drawVectors;
protected override void Initialize()
{
    myVectors = new Vector2[9];
    drawVectors = new Vector2[9];

    base.Initialize();
}
```

3. After loading the sprite, calculate the positions of the unrotated group of sprites based on the sprite's size.

C#

```
private Texture2D SpriteTexture;
private Vector2 origin;
private Vector2 screenpos;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    SpriteTexture = Content.Load<Texture2D>("ship");
    origin.X = SpriteTexture.Width / 2;
    origin.Y = SpriteTexture.Height / 2;
    Viewport viewport = graphics.GraphicsDevice.Viewport;
    screenpos.X = viewport.Width / 2;
    screenpos.Y = viewport.Height / 2;
}
```

4. In your [Update](#) method, copy the unrotated vectors and determine the screen position around which all the sprites will rotate.

C#

```
private float RotationAngle = 0f;
private bool batchAutoRotate = false;
private Matrix rotationMatrix = Matrix.Identity;
protected override void Update(GameTime gameTime)
```

```

{
    ...
    float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;

    RotationAngle += elapsed;
    float circle = MathHelper.Pi * 2;
    RotationAngle = RotationAngle % circle;

    // Copy and rotate the sprite positions.
    drawVectors = (Vector2[])myVectors.Clone();

    if (!batchAutoRotate)
        RotatePoints(ref screenpos, RotationAngle, ref drawVectors);
    else
        UpdateMatrix(ref screenpos, RotationAngle);

    base.Update(gameTime);
}

```

Now you need to transform each vector using a rotation matrix created for the rotation angle.

5. To rotate around the origin, transform each vector relative to the origin by subtracting the origin vector.
6. Add the origin vector to the transformed vector to create the final rotated vector.

C#

```

private static void RotatePoints(ref Vector2 origin, float radians,
    ref Vector2[] Vectors)
{
    Matrix myRotationMatrix = Matrix.CreateRotationZ(radians);

    for (int i = 0; i < 9; i++)
    {
        // Rotate relative to origin.
        Vector2 rotatedVector =
            Vector2.Transform(Vectors[i] - origin, myRotationMatrix);

        // Add origin to get final location.
        Vectors[i] = rotatedVector + origin;
    }
}

```

7. Draw each sprite using the rotated vectors as screen locations.

C#

```

private void DrawPoints()
{
    // Draw using manually rotated vectors
    spriteBatch.Begin();
    for (int i = 0; i < drawVectors.Length; i++)
        spriteBatch.Draw(SpriteTexture, drawVectors[i], null,
            Color.White, RotationAngle, origin, 1.0f,
            SpriteEffects.None, 0f);
    spriteBatch.End();
}

```

8. When all the sprites have been drawn, call `End`.

To draw a rotated group of sprites on screen using `SpriteBatch` transformation

1. Follow the steps of [How To: Draw a Sprite](#).
2. Create one set of [Vector2](#) objects that represents the unrotated positions of the sprites, and one set to hold the rotated values.

C#

```
private Vector2[] myVectors;
private Vector2[] drawVectors;
protected override void Initialize()
{
    myVectors = new Vector2[9];
    drawVectors = new Vector2[9];

    base.Initialize();
}
```

3. After loading the sprite, calculate the positions of the unrotated group of sprites based on the sprite's size.

C#

```
private Texture2D SpriteTexture;
private Vector2 origin;
private Vector2 screenpos;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    SpriteTexture = Content.Load<Texture2D>("ship");
    origin.X = SpriteTexture.Width / 2;
    origin.Y = SpriteTexture.Height / 2;
    Viewport viewport = graphics.GraphicsDevice.Viewport;
    screenpos.X = viewport.Width / 2;
    screenpos.Y = viewport.Height / 2;
}
```

4. In your [Update](#) method, copy the unrotated vectors and determine the screen position around which all the sprites will rotate.

C#

```
private float RotationAngle = 0f;
private bool batchAutoRotate = false;
private Matrix rotationMatrix = Matrix.Identity;
protected override void Update(GameTime gameTime)
{
    ...
    float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;

    RotationAngle += elapsed;
    float circle = MathHelper.Pi * 2;
    RotationAngle = RotationAngle % circle;

    // Copy and rotate the sprite positions.
    drawVectors = (Vector2[])myVectors.Clone();

    if (!batchAutoRotate)
        RotatePoints(ref screenpos, RotationAngle, ref drawVectors);
    else
        UpdateMatrix(ref screenpos, RotationAngle);
}
```

```
    base.Update(gameTime);  
}
```

5. Create a rotation matrix for [SpriteBatch](#) to use during [Draw](#).
6. To rotate around a point that is not the upper-left corner of the screen, first subtract the origin of your rotation using a translation matrix.
7. Multiply that translation matrix by the rotation matrix.
8. Multiply the result by a translation matrix adding the origin of your rotation.

This is because a rotation matrix must rotate around (0,0,0).

C#

```
private void UpdateMatrix(ref Vector2 origin, float radians)  
{  
    // Translate sprites to center around screen (0,0), rotate them, and  
    // translate them back to their original positions  
    Vector3 matrixorigin = new Vector3(origin, 0);  
    rotationMatrix = Matrix.CreateTranslation(-matrixorigin) *  
        Matrix.CreateRotationZ(radians) *  
        Matrix.CreateTranslation(matrixorigin);  
}
```

9. In your [Draw](#) method, call [SpriteBatch.Begin](#), specifying your rotation matrix as the fourth parameter.

C#

```
private void DrawMatrix()  
{  
    // Draw using a rotation matrix with SpriteBatch  
    spriteBatch.Begin(SpriteBlendMode.AlphaBlend,  
        SpriteSortMode.Deferred, SaveStateMode.None, rotationMatrix);  
    for (int j = 0; j < myVectors.Length; j++)  
        spriteBatch.Draw(SpriteTexture, myVectors[j], null, Color.White,  
            0, origin, 1.0f, SpriteEffects.None, 0f);  
    spriteBatch.End();  
}
```

10. When all the sprites have been drawn, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

[CreateRotationZ](#)

How To: Scale a Sprite

Demonstrates how to scale a sprite using a uniform scale, nonuniform scale, or a destination rectangle.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ScaleSprite_Sample.zip.](#)

Drawing a Scaled Sprite

To draw a scaled sprite with a uniform scale

1. Follow the procedures of [How To: Draw a Sprite](#).
2. In your **Update** method, determine how your sprite will be scaled.

The normal size of the sprite is multiplied by the scale specified. For example, a value of 1.0 draws the sprite full size, where 0.5 will draw it half-sized and 2.0 will draw it at twice its original size.

C#

```
protected float scale = 1f;
protected override void Update(GameTime gameTime)
{
    ...

    // The time since Update was called last.
    float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;

    // TODO: Add your game logic here.
    scale += elapsed;
    scale = scale % 6;

    base.Update(gameTime);
}
```

3. When drawing the sprite, specify the scale of the sprite as a parameter to draw.

Specifying a floating-point scale parameter scales the sprite evenly in both the x and y directions.

C#

```
protected virtual void DrawForeground( SpriteBatch batch )
{
    Rectangle safeArea = GetTitleSafeArea( .8f );
    Vector2 position = new Vector2( safeArea.X, safeArea.Y );
    batch.Begin();
    batch.Draw( SpriteTexture, position, null,
        Color.White, 0f, Vector2.Zero, scale, SpriteEffects.None, 0f );
    batch.End();
}
```

4. When all the sprites have been drawn, call **End** on your [SpriteBatch](#) object.

To draw a scaled sprite using a nonuniform scale

1. Follow the procedures of [How To: Draw a Sprite](#).
2. In your **Update** method, determine how your sprite will be scaled along each axis, and store those values in a [Vector2](#) object.

C#

```
protected Vector2 nonuniformscale = Vector2.One;
protected override void Update(GameTime gameTime)
{
    base.Update(gameTime);
    float basescale = nonuniformscale.Y;
    basescale += (float)gameTime.ElapsedGameTime.TotalSeconds;
    basescale = basescale % 6;
    nonuniformscale.Y = basescale;
    nonuniformscale.X = basescale * .8f;
}
```

3. When drawing the sprite, specify the scale of the sprite using the [Vector2](#) object that you updated earlier.

Specifying a [Vector2](#) scales the sprite independently in both the x and y directions.

C#

```
protected override void DrawForeground(SpriteBatch batch)
{
    Rectangle safeArea = GetTitleSafeArea(.8f);
    Vector2 position = new Vector2(safeArea.X, safeArea.Y);
    batch.Begin();
    batch.Draw(SpriteTexture, position, null, Color.White, 0, Vector2.Zero,
        nonuniformscale, SpriteEffects.None, 0);
    batch.End();
}
```

4. When all of the sprites have been drawn, call [End](#) on your [SpriteBatch](#) object.

To draw a scaled sprite using a destination rectangle

1. Follow the procedures of [How To: Draw a Sprite](#).
2. In your **Update** method, construct a rectangle that defines where on screen the sprite will be drawn.

This rectangle does not need to be the same shape or size as the original sprite. Each dimension of the sprite is scaled independently to fit the destination rectangle.

C#

```
protected Rectangle destrect;
protected override void Update(GameTime gameTime)
{
    destrect = new Rectangle();
    Rectangle safeArea = GetTitleSafeArea(.8f);
    destrect.X = safeArea.X;
    destrect.Y = safeArea.Y;
    destrect.Width = (int)scale * 100;
    destrect.Height = (int)scale * 80;
    base.Update(gameTime);
}
```

3. When drawing the sprite, specify the destination rectangle as a parameter to [Draw](#).

The sprite will be drawn, filling the destination rectangle.

C#

```
protected override void DrawForeground(SpriteBatch batch)
{
    batch.Begin();
```



```
    batch.Draw(SpriteTexture, destrect, Color.White);  
    batch.End();  
}
```

4. When all of the sprites have been drawn, call [End](#) on your [SpriteBatch](#) object.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Tile a Sprite

Demonstrates how to draw a sprite repeatedly in the x and y directions in one [Draw](#) call.

This sample takes advantage of the texture addressing modes of the graphics card to duplicate a texture across the area defined for drawing by [SpriteBatch.Draw](#). Using [SpriteSortMode.Immediate](#) allows us to set the [TextureAddressMode](#) directly and enable wrapping. Other address modes (such as mirroring) can create interesting results.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download TiledSprites_Sample.zip.](#)

Drawing a Tiled Sprite

To draw a tiled sprite on screen

1. Follow the procedures of [How To: Draw a Sprite](#).
2. In the [Draw](#) method, create a [Rectangle](#), defining how many pixels to fill with your sprite's image.

The total width of the [Rectangle](#) should be the number of times to tile your sprite in the x direction multiplied by the width of the sprite. The total height of the [Rectangle](#) should be the number of times to tile your sprite in the y direction multiplied by the height of the sprite.

C#

```
int TilesX = 2;
int TilesY = 2;
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    Rectangle source = new Rectangle(0, 0, xna.Width * TilesX,
        xna.Height * TilesY);
```

3. Call [SpriteBatch.Begin](#), and then select [SpriteSortMode.Immediate](#).
4. Set the [TextureAddressMode](#) of the first [SamplerState](#) to [TextureAddressMode.Wrap](#) for both the u and v directions.

C#

```
spriteBatch.Begin(SpriteBlendMode.AlphaBlend,
    SpriteSortMode.Immediate, SaveStateMode.None);
GraphicsDevice.SamplerStates[0].AddressU = TextureAddressMode.Wrap;
GraphicsDevice.SamplerStates[0].AddressV = TextureAddressMode.Wrap;
```

5. Call [SpriteBatch.Draw](#) with the sprite to tile, the starting position of the sprite, the source rectangle you created, and other relevant parameters.

C#

```
Vector2 pos = new Vector2(50);
spriteBatch.Draw(xna, pos, source, Color.White, 0,
    Vector2.Zero, 0.5f, SpriteEffects.None, 1.0f);
```

6. When all of the sprites have been drawn, call [End](#) on your [SpriteBatch](#) object.

C#

```
spriteBatch.End();

base.Draw(gameTime);
}
```

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

[What Is Texture Mapping?](#)

Reference

[SpriteBatch](#)

[Draw](#)

[SpriteSortMode](#)

[Texture2D](#)

How To: Tint a Sprite

Demonstrates how to tint a sprite using a [Color](#) value.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download TintSprite_Sample.zip](#).

Drawing a Tinted Sprite

To draw a tinted sprite on screen

1. Follow the procedures of [How To: Draw a Sprite](#).
2. In the [Update](#) method, determine how to tint the sprite.

In this example, we take the value of the game pad thumbsticks and determine Red, Green, Blue, and Alpha values to apply to the sprite.

C#

```
protected Color tint;
protected override void Update(GameTime gameTime)
{
    ...
    GamePadState input = GamePad.GetState(PlayerIndex.One);
    tint = new Color(GetColor(input.ThumbSticks.Left.X),
        GetColor(input.ThumbSticks.Left.Y),
        GetColor(input.ThumbSticks.Right.X),
        GetColor(input.ThumbSticks.Right.Y));

    base.Update(gameTime);
}
```

3. In the [Draw](#) method, pass the color value created in [Update](#) to [SpriteBatch.Draw](#).

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();
    spriteBatch.Draw(SpriteTexture, position, tint);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

4. When all of the sprites have been drawn, call [End](#) on your [SpriteBatch](#) object.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

How To: Scale Sprites Based On Screen Size

Demonstrates how to scale sprites using a matrix that is created based on the viewport width.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ScaleScreen_Sample.zip.](#)

Scaling Sprites Based on Screen Size

To scale sprites based on screen size

1. Use the [PreferredBackBufferHeight](#) and [PreferredBackBufferWidth](#) properties of [GraphicsDeviceManager](#) during your game's [Initialize](#) to set the default screen size of your game.
2. In your [LoadContent](#) method, use [Matrix.CreateScale](#) to create a scaling matrix.

This matrix is recreated any time the resolution of the [GraphicsDevice](#) changes.

Because you are scaling sprites, you should use only the x and y parameters to create the scaling matrix. Scaling the depth of sprites can result in their depth shifting above 1.0. If that happens, they will not render.

C#

```
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    ...

    // Default resolution is 800x600; scale sprites up or down based on
    // current viewport
    float screenscale =
        (float) graphics.GraphicsDevice.Viewport.Width / 800f;
    // Create the scale transform for Draw.
    // Do not scale the sprite depth (Z=1).
    SpriteScale = Matrix.CreateScale( screenscale, screenscale, 1 );
}
```

3. In your [Update](#) method, determine whether the game needs to change screen resolution.

This example uses game pad buttons to switch between two resolutions.

C#

```
protected override void Update(GameTime gameTime)
{
    ...
    // Change the resolution dynamically based on input
    if (GamePad.GetState(PlayerIndex.One).Buttons.A ==
        ButtonState.Pressed)
    {
        graphics.PreferredBackBufferHeight = 768;
        graphics.PreferredBackBufferWidth = 1024;
        graphics.ApplyChanges();
    }
    if (GamePad.GetState(PlayerIndex.One).Buttons.B ==
        ButtonState.Pressed)
    {
```

```
        graphics.PreferredBackBufferHeight = 600;
        graphics.PreferredBackBufferWidth = 800;
        graphics.ApplyChanges();
    }
    base.Update(gameTime);
}
```

4. In your `Draw` method, call `SpriteBatch.Begin`, passing the scaling matrix created in `LoadContent`.
5. Draw your scene normally, then call `SpriteBatch.End`.

All of the sprites you draw will be scaled according to the matrix.

C#

```
protected override void Draw(GameTime gameTime)
{
    ...
    // Initialize the batch with the scaling matrix
    spriteBatch.Begin(SpriteBlendMode.AlphaBlend,
        SpriteSortMode.Deferred, SaveStateMode.None, SpriteScale);
    // Draw a sprite at each corner
    for (int i = 0; i < spritepos.Length; i++)
    {
        spriteBatch.Draw(square, spritepos[i], null, Color.White,
            rotation, origin, scale, SpriteEffects.None, depth);
    }
    spriteBatch.End();
    base.Draw(gameTime);
}
```

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[Draw](#)

[Texture2D](#)

[Matrix.CreateScale](#)

How To: Draw Point Sprites

Demonstrates how to create and draw point sprites.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download PointSprites_Sample.zip.](#)

Using Point Sprites

To create a point sprite shader

- Verify that the shader for point sprites has a pixel shader that accepts a texture coordinate.

On Xbox 360, these are specified by the special SPRITETEXCOORD semantic. On Windows-based computers, any texture coordinate such as TEXCOORD0 will work. The pixel shader returns the color of the texel at the coordinate in the texture that is used to draw the sprite.

```
uniform extern texture SpriteTexture;
struct PS_INPUT
{
    #ifdef XBOX
        float2 TexCoord : SPRITETEXCOORD;
    #else
        float2 TexCoord : TEXCOORD0;
    #endif
};
sampler Sampler = sampler_state
{
    Texture = <SpriteTexture>;
};
float4 PixelShader(PS_INPUT input) : COLOR0
{
    float2 texCoord;

    texCoord = input.TexCoord.xy;

    return tex2D(Sampler, texCoord);
}
```

A vertex shader for point sprites only needs to return a POSITION0 for the vertex, transformed by the world-view-projection matrix.

```
uniform extern float4x4 WVPMatrix;
float4 VertexShader(float4 pos : POSITION0) : POSITION0
{
    return mul(pos, WVPMatrix);
}
```

To draw a group of point sprites

1. In [LoadContent](#), load an [Effect](#) object for your point sprites, and set the texture the point sprites will draw.

This example also creates a random set of point sprites to draw, using an array of [VertexPositionColor](#) elements.

C#


```

Effect pointSpritesEffect;
VertexPositionColor[] spriteArray;
VertexDeclaration vertexPosColDecl;
Random rand;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    pointSpritesEffect = Content.Load<Effect>("pointsprites");
    pointSpritesEffect.Parameters["SpriteTexture"].SetValue(
        Content.Load<Texture2D>("fire"));
    spriteArray = new VertexPositionColor[200];
    vertexPosColDecl = new VertexDeclaration(graphics.GraphicsDevice,
        VertexPositionColor.VertexElements);
    rand = new Random();
    for (int i = 0; i < spriteArray.Length; i++)
    {
        spriteArray[i].Position = new Vector3(rand.Next(100) / 10f,
            rand.Next(100) / 10f, rand.Next(100) / 10f);
        spriteArray[i].Color = Color.WhiteSmoke;
    }
}

```

- In [Draw](#), set the proper render states on the [GraphicsDevice](#).
- Set [PointSizeEnable](#) to **true** and [DepthBufferWriteEnable](#) to **false**.
- If your shader does not set the PSIZE semantic itself, set a [PointSize](#) here.

The [PointSize](#) is the width of the point sprite on screen, in pixels. Setting the PSIZE semantic in the shader allows the shader to vary the size of each point. Using the [PointSize](#) render state instead will cause each point to be rendered as the same size.

- Set up additive blending by setting [AlphaBlendEnable](#) to **true**, and the [SourceBlend](#) and [DestinationBlend](#) to [Blend.One](#).

C#

```

graphics.GraphicsDevice.RenderState.PointSpriteEnable = true;
graphics.GraphicsDevice.RenderState.PointSize = 64.0f;
graphics.GraphicsDevice.RenderState.AlphaBlendEnable = true;
graphics.GraphicsDevice.RenderState.SourceBlend = Blend.SourceAlpha;
graphics.GraphicsDevice.RenderState.DestinationBlend = Blend.One;
graphics.GraphicsDevice.RenderState.DepthBufferWriteEnable = false;

```

- Set the [VertexDeclaration](#) on the [GraphicsDevice](#) to the type of vertex you will be drawing (in this case, [VertexPositionColor](#)).

C#

```

graphics.GraphicsDevice.VertexDeclaration
    = vertexPosColDecl;

```

- Call [SetValue](#) for any draw-time parameters to your [Effect](#), such as the world-view-projection matrix.

C#

```

Matrix WVPMatrix = Matrix.Identity * viewMatrix * projectionMatrix;
pointSpritesEffect.Parameters["WVPMatrix"].SetValue(WVPMatrix);

```

- Call [Begin](#) on your [Effect](#), and for each pass in the effect, call [DrawUserPrimitives](#) for the point sprites.

C#

```
pointSpritesEffect.Begin();
foreach (EffectPass pass in
    pointSpritesEffect.CurrentTechnique.Passes)
{
    pass.Begin();
    graphics.GraphicsDevice.DrawUserPrimitives
        <VertexPositionColor>(
        PrimitiveType.PointList,
        spriteArray,
        0,
        spriteArray.Length);
    pass.End();
}
pointSpritesEffect.End();
```

9. Reset your render states for drawing any subsequent objects.

C#

```
graphics.GraphicsDevice.RenderState.PointSpriteEnable = false;
graphics.GraphicsDevice.RenderState.DepthBufferWriteEnable = true;
graphics.GraphicsDevice.RenderState.SourceBlend = Blend.SourceAlpha;
graphics.GraphicsDevice.RenderState.DestinationBlend =
    Blend.InverseSourceAlpha;
```

See Also

Concepts

[HLSL Input Semantics \(Xbox 360\)](#)

Reference

[PointSpriteEnable](#)

[PointSize](#)

[DrawUserPrimitives](#)

How To: Apply a Pixel Shader to Sprites

Demonstrates how to apply a pixel shader to sprites.

This example uses an offscreen render target to render a number of sprites, then applies a pixel shader to the full scene. This allows the pixel shader to shade the entire screen instead of each individual sprite separately.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download SpritePixelShader_Sample.zip.](#)

Applying a Pixel Shader to a Sprite

To create a sprite pixel shader

- Create a sampler object with a **Texture** member.

This is the texture that will be changed by the pixel shader. Although we declare the texture as an external parameter, actually the [SpriteBatch](#) populates the texture when it draws to the screen.

```
uniform extern texture ScreenTexture;

sampler ScreenS = sampler_state
{
    Texture = <ScreenTexture>;
};
```

A pixel shader is only required to return a color for the current pixel. This pixel shader uses the **tex2D** function to pick a pixel on the sampler to display. Because this is a distortion shader, the pixel chosen is not necessarily the "current" pixel.

```
float4 PixelShader(float2 texCoord: TEXCOORD0) : COLOR
{
    ...
    // pick a pixel on the screen for this pixel, based on
    // the calculated offset and direction
    float4 color = tex2D(ScreenS, texCoord+(sinoffset*sinsign));

    return color;
}
```

To apply a pixel shader to sprites

1. In [LoadContent](#), create a [SpriteBatch](#), load the [Effect](#) containing your pixel shader, and load any sprites you want to draw.
2. In the section of [LoadContent](#) for loading manual resources, create a [RenderTarget2D](#) for offscreen rendering, and create a [Texture2D](#) using the same settings.

The [Texture2D](#) will hold the scene that we later pass to the pixel shader.

C#

```
RenderTarget2D ShaderRenderTarget;
Texture2D ShaderTexture;
Effect ripple;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    grid = Content.Load<Texture2D>("grid");
}
```

```

    ripple = Content.Load<Effect>("ripple");
    ...

    ShaderRenderTarget = CloneRenderTarget(GraphicsDevice, 1);
    ShaderTexture = new Texture2D(GraphicsDevice,
        ShaderRenderTarget.Width, ShaderRenderTarget.Height, 1,
        TextureUsage.None, ShaderRenderTarget.Format);

    ...
}

```

3. In `Draw`, cache the current render target using `GetRenderTarget`, then switch to an offscreen render target using `SetRenderTarget`.
4. `Clear` the offscreen render target, and then draw all the sprites that comprise the scene.
5. Resolve the render target with `ResolveRenderTarget`, and retrieve the scene with `GetTexture`.
6. Reset the render target to the previous render target (normally the back buffer).

C#

```

// Change to our offscreen render target.
RenderTarget2D temp =
    (RenderTarget2D)GraphicsDevice.GetRenderTarget(0);
GraphicsDevice.SetRenderTarget(0, ShaderRenderTarget);

GraphicsDevice.Clear(Color.Black);
// Render a simple scene.
spriteBatch.Begin();
TileSprite(spriteBatch, grid);
spriteBatch.End();

// Change back to the back buffer, and get our scene
// as a texture.
GraphicsDevice.SetRenderTarget(0, temp);
ShaderTexture = ShaderRenderTarget.GetTexture();

```

7. Using the `Texture2D` containing the scene to shade, call `Begin` on your `SpriteBatch`, specifying `SpriteSortMode.Immediate`.
 8. Call `Begin` on the `Effect`, and `Begin` on the pass containing your pixel shader.
 9. Call `Draw` on the `SpriteBatch`.
- The pixel shader you specified will be used to render the sprite.
10. Call `End` on the `SpriteBatch`, the `EffectPass`, and the `Effect`.

C#

```

// Use Immediate mode and our effect to draw the scene
// again, using our pixel shader.
spriteBatch.Begin(SpriteBlendMode.None, SpriteSortMode.Immediate,
    SaveStateMode.None);
ripple.Begin();
ripple.CurrentTechnique.Passes[0].Begin();
spriteBatch.Draw(ShaderTexture, Vector2.Zero, Color.White);
spriteBatch.End();
ripple.CurrentTechnique.Passes[0].End();
ripple.End();

```

[2D Graphics Overview](#)

[What Is a Render Target?](#)

Reference

[Draw](#)

[SpriteSortMode.Immediate](#)

[Effect](#)

[RenderTarget2D](#)

How To: Draw a Sprite Over a Model

Demonstrates how to draw a sprite so that it obscures a model. In this example, we are drawing an animated sprite representing an explosion over the current screen position of a 3D model.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ViewportProjection_Sample.zip](#).

For this sample, the camera is a standard arc ball camera, implemented by camera.cs. The 3D model file is a simple ring, implemented by ring16b.x. The animated explosion sprite is implemented by explosion.dds. These files can be found in the complete sample. See [How To: Animate a Sprite](#) for an example of the AnimatedTexture class.

Drawing a Sprite Over a Model

To draw a sprite over a model

1. In your [Update](#) method, handle the input to move your camera, then call **UpdateFrame** on the **AnimatedTexture**.

C#

```
protected override void Update(GameTime gameTime)
{
    GamePadState PlayerOne = GamePad.GetState(PlayerIndex.One);

    // Move the camera using thumbsticks
    MoveCamera(PlayerOne);

    // Start or stop the animated sprite using buttons
    if (PlayerOne.Buttons.A == ButtonState.Pressed)
        explosion.Play();
    if (PlayerOne.Buttons.B == ButtonState.Pressed)
        explosion.Stop();

    // Update the animated sprite
    explosion.UpdateFrame((float)gameTime.ElapsedGameTime.TotalSeconds);
}
```

2. Use [CreateMerged](#) to create a [BoundingSphere](#) that contains all the [BoundingSphere](#) values for each [ModelMesh](#) in the [Model](#).
3. Use [Viewport.Project](#) to find the centerpoint of that sphere, which is the center of the model in screen coordinates.

C#

```
// Create a total bounding sphere for the mesh
BoundingSphere totalbounds = new BoundingSphere();
foreach (ModelMesh mesh in Ring.Meshes)
{
    totalbounds = BoundingSphere.CreateMerged(totalbounds,
        mesh.BoundingSphere);
}

// Project the center of the 3D object to the screen, and center the
// sprite there
Vector3 center = GraphicsDevice.Viewport.Project(totalbounds.Center,
    projectionMatrix, Camera1.ViewMatrix, Matrix.Identity);
explosionpos.X = center.X;
explosionpos.Y = center.Y;
```

4. Take the [BoundingSphere](#) for the model and use that to create a [BoundingBox](#) with [CreateFromSphere](#).
5. Find the corner of the box that is farthest from the center using [Project](#), and use that to scale the sprite appropriately.

C#

```
// Create a bounding box from the bounding sphere,
// and find the corner that is farthest away from
// the center using Project
BoundingBox extents = BoundingBox.CreateFromSphere(totalbounds);
float maxdistance = 0;
float distance;
Vector3 screencorner;
foreach (Vector3 corner in extents.GetCorners())
{
    screencorner = GraphicsDevice.Viewport.Project(corner,
        projectionMatrix, Camera1.ViewMatrix, Matrix.Identity);
    distance = Vector3.Distance(screencorner, center);
    if (distance > maxdistance)
        maxdistance = distance;
}

// Scale the sprite using the two points (the sprite is
// 75 pixels square)
explosion.Scale = maxdistance / 75;

base.Update(gameTime);
}
```

6. In your [Draw](#) method, draw the [Model](#) normally, then draw the animated sprite using the position calculated in [Update](#).

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    //Draw the model, a model can have multiple meshes, so loop
    foreach (ModelMesh mesh in Ring.Meshes)
    {
        //This is where the mesh orientation is set, as well as
        //our camera and projection
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.EnableDefaultLighting();
            effect.World = Matrix.Identity *
                Matrix.CreateRotationY( RingRotation ) *
                Matrix.CreateTranslation( RingPosition );
            effect.View = Camera1.ViewMatrix;
            effect.Projection = projectionMatrix;
        }
        //Draw the mesh, will use the effects set above.
        mesh.Draw();
    }

    // Draw the sprite over the 3D object
    spriteBatch.Begin( SpriteBlendMode.AlphaBlend,
        SpriteSortMode.Deferred, SaveStateMode.SaveState );
    explosion.DrawFrame( spriteBatch, explosionpos );
    spriteBatch.End();
}
```

```
    base.Draw( gameTime );  
}
```

See Also

Tasks

[How To: Animate a Sprite](#)

[How To: Render a Model](#)

Concepts

[2D Graphics Overview](#)

[What Is Color Blending?](#)

Reference

[Project](#)

[BoundingSphere](#)

[BoundingBox](#)

How To: Draw Text

Demonstrates how to import a [SpriteFont](#) into a project and draw text using [DrawString](#).

Note

As with most types of software, font files are licensed rather than sold. Font licenses vary from vendor to vendor, but most do not allow redistribution of the fonts. That includes redistribution of reproductions such as bitmaps containing the rasterized character set. This is true of many of the licenses covering fonts that Microsoft supplies with applications and the Windows operating systems. Therefore, be careful to ensure that you have the required license rights to redistribute any font you include as a bitmap containing the rasterized character set in your game!

Note

The following list of fonts are installed by XNA Game Studio and are redistributable:

- Kooten.ttf
- Linds.ttf
- Miramo.ttf
- Bold Miramob.ttf
- Peric.ttf
- Pericl.ttf
- Pesca.ttf
- Pescab.ttf

These OpenType fonts, created by Ascender Corporation and licensed by Microsoft, are free for you to use in your XNA Game Studio game. You may redistribute these fonts in their original format as part of your game. These fonts were previously available only on the XNA Creators Club Online website.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download FontDemo_Sample.zip](#).

Adding a Sprite Font and Drawing Text

To add a sprite font

1. Right-click your Content project in Solution Explorer, click **Add**, and then click **New Item**.
2. From the **Add New Item** dialog box, click **Sprite Font**.

You may find it convenient at this point to change the name of the new file from "SpriteFont1" to the friendly name of the font you intend to load (leaving the file extension as ".spritefont"). The friendly name is the name that identifies the font once it is installed on your computer. For example, "Courier New" or "Times New Roman."

XNA Game Studio creates a new .spritefont file for your font and opens it.

3. If you did not name the new file with the font's friendly name, type the friendly name of the font to load into the FontName element.

Again, this is not the name of a font file, but rather the name that identifies the font once it is installed on your computer. You can use the Fonts folder in the **Control Panel** to see the names of fonts installed on your system, and to install new ones as well. The content pipeline supports the same fonts as the [System.Drawing.Font](#) class, including TrueType fonts, but not bitmap (.fon) fonts. You may find it convenient to save the new .spritefont file using this friendly name.

4. If necessary, change the **Size** entry to the point size you desire for your font.
5. If necessary, change the **Style** entry to the style of font to import.

You can specify **Regular**, **Bold**, **Italic**, or **Bold, Italic**. The **Style** entry is case sensitive.

6. Specify the character regions to import for this font.

Character regions specify which characters in the font are rendered by the [SpriteFont](#). You can specify the start and end of the region by using the characters themselves, or by using their decimal values with a `&#` prefix. The default character region includes all the characters between the space and tilde characters, inclusive.

To draw text on screen

1. Add a [Sprite Font](#) to your project as described above.
2. Create a [SpriteFont](#) object to encapsulate the imported font.
3. Create a [SpriteBatch](#) object for drawing the font onscreen.
4. In your [LoadContent](#) method, call [ContentManager.Load](#), specifying the [SpriteFont](#) class and the asset name of the imported font.
5. Create your [SpriteBatch](#) object, passing the current [GraphicsDevice](#).

C#

```
SpriteFont Font1;
Vector2 FontPos;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    Font1 = Content.Load<SpriteFont>("Courier New");

    // TODO: Load your game content here
    FontPos = new Vector2(graphics.GraphicsDevice.Viewport.Width / 2,
        graphics.GraphicsDevice.Viewport.Height / 2);
}
```

6. In your [Draw](#) method, call [Begin](#) on the [SpriteBatch](#) object.

If you select your own [SpriteBlendMode](#), [SpriteBlendMode.AlphaBlend](#) is recommended.

7. If necessary, determine the origin of your text.

If you want to draw your text centered on a point, you can find the center of the text by calling [MeasureString](#) and dividing the returned vector by 2.

8. Call [DrawString](#) to draw your output text, specifying the [SpriteFont](#) object for the font you want to use.

All other parameters of [DrawString](#) produce the same effects as a call to [SpriteBatch.Draw](#).

9. When you have drawn all your text, call [SpriteBatch.End](#).

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();

    // Draw Hello World
    string output = "Hello World";

    // Find the center of the string
    Vector2 FontOrigin = Font1.MeasureString( output ) / 2;
    // Draw the string
    spriteBatch.DrawString( Font1, output, FontPos, Color.LightGreen,
        0, FontOrigin, 1.0f, SpriteEffects.None, 0.5f );

    spriteBatch.End();
    base.Draw( gameTime );
}
```

```
}
```

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch](#)

[DrawString](#)

[SpriteFont](#)

[ContentManager.Load](#)

3D Graphics

Provides an overview of the 3D Graphics classes as well as tutorials to demonstrate low-level 3D rendering.

In This Section

[3D Graphics Overview](#)

Provides an overview of 3D graphics. The [Microsoft.Xna.Framework.Graphics](#) namespace contains classes to use the graphics device to load and render resources and to apply effects to vertices and pixels.

[Viewports and Frustums](#)

Provides overview of viewports and frustums as related to 3D graphics concepts.

[Dynamic Vertex Buffers in XNA](#)

Describes several techniques for implementing dynamic vertex buffers in an XNA Framework game.

["How To" Articles](#)

Describes step-by-step procedures for certain 3D graphics techniques.

3D Graphics Overview

Provides an overview of 3D graphics. The [Microsoft.Xna.Framework.Graphics](#) namespace contains classes to use the graphics device to load and render resources and to apply effects to vertices and pixels. This overview covers the following topics as they apply to 3D graphics.

- [Getting Started with 3D](#)
- [The Graphics Device](#)
- [Resources](#)
- [Shaders and Effects](#)

Getting Started with 3D

In the examples in this section, you must provide the following three items for even the most simple 3D graphics application:

- A set of world, view, and projection transformation matrices
- A vertex buffer
- An effect that applies the world, view, and projection transformation matrices to the vertex data

This list of items enables you to render your first 3D content in an XNA application.

As you become comfortable with these ideas, you will want to learn more about the following:

- Manipulating the vertices you created
- Creating your own effects
- Applying textures
- Improving the performance of your application through such techniques as converting your vertex buffers to indexed vertex streams

The XNA Framework uses a shader-driven programmable pipeline. It requires a graphics card capable of at least Shader Model 1.1. Shader Model 2.0 is recommended. In the programmable pipeline, shaders and effects are used to apply transformations, textures, lighting, and materials. However, you do not need to write a custom shader for your first 3D XNA applications. The XNA Framework provides a class called [BasicEffect](#) that encapsulates most of these common operations.

The Graphics Device

When you create a game with XNA Game Studio, the XNA application model initializes a graphics device for you. You can use the [GraphicsDeviceManager.GraphicsDevice](#) property to gain access to the graphics device. To determine the capabilities of the graphics device, use the [GraphicsDevice.GraphicsDeviceCapabilities](#) property.

Graphics Device Initialization

The [GraphicsDeviceManager](#) initializes the [GraphicsDevice](#) before you call [Game.Initialize](#). Before you call [Initialize](#), there are three ways to change the [GraphicsDevice](#) settings:

1. Set the appropriate properties (e.g. [PreferredBackBufferHeight](#), [PreferredBackBufferWidth](#)) on the [GraphicsDeviceManager](#) in your game's constructor.
2. Handle the [PreparingDeviceSettings](#) event on the [GraphicsDeviceManager](#), and change the [PreparingDeviceSettingsEventArgs.GraphicsDeviceInformation.PresentationParameters](#) member properties.

Any changes made to the [PreparingDeviceSettingsEventArgs](#) will override the [GraphicsDeviceManager](#) preferred settings.

3. Handle the [DeviceCreated](#) event on the [GraphicsDeviceManager](#), and change the [PresentationParameters](#) of the [GraphicsDevice](#) directly.

When you call [Game.Initialize](#), [GraphicsDeviceManager](#) creates and configures [GraphicsDevice](#). You can safely access [GraphicsDevice](#) settings such as the backbuffer, depth/stencil buffer, viewport, and render states in [Initialize](#).

After you call [Game.Initialize](#), changes to the [PresentationParameters](#) of the [GraphicsDevice](#) will not take effect until you call [GraphicsDeviceManager.ApplyChanges](#). Other changes, such as render states, will take effect immediately.

Render States and the Graphics Device

Render states are an important part of the graphics device. They can affect game rendering significantly. When you first create a graphics device, the render states are set to their default values. For more information, see [RenderStates](#).

Commonly, the value of a render state is changed only if it is manually modified. For example, the following code sample sets the cull mode of the current graphics device to `None`. This causes all triangle faces to be drawn and differs from the default value of `CullMode.CounterClockwise`.

```
renderState.CullMode = CullMode.None;
```

However, there are cases where render states can change automatically. A common example is the rendering of sprites and 3D objects on the same graphics device. In this case, the [SpriteBatch](#) object changes various render states when you call [End](#). If you try any 3D rendering after this step, the results can be unpredictable.

For this reason, you should restore several key render states to their former settings before you try to render any 3D objects. The following code demonstrates this approach.

```
GraphicsDevice.RenderState.DepthBufferEnable = true;  
GraphicsDevice.RenderState.AlphaBlendEnable = false;  
GraphicsDevice.RenderState.AlphaTestEnable = false;
```

In addition, you should set the following render states, depending on the type of 3D content you are rendering.

```
GraphicsDevice.SamplerStates[0].AddressU = TextureAddressMode.Wrap;  
GraphicsDevice.SamplerStates[0].AddressV = TextureAddressMode.Wrap;
```

It is also possible to save the render state of the graphics device before rendering sprites by passing [SaveStateMode.SaveState](#) your [SpriteBatch.Begin](#) method call. However, this is not recommended because it is time intensive, and can slow down the rendering process.

Resources

A resource is a collection of data stored in memory that can be accessed by the CPU or GPU. Types of resources that an application might use include render targets, vertex buffers, index buffers, and textures.

Based on the resource management mode that was used when a resource is created, it should be reloaded when the device is reset. For more information, see [How To: Load Content](#).

Vertex and Index Buffers

A vertex buffer contains a list of 3D vertices to be streamed to the graphics device. Each vertex in a vertex buffer may contain data about not only the 3D coordinate of the vertex, but also other information describing the vertex, such as the vertex normal, color, or texture coordinate. The XNA Framework contains several classes to describe common vertex declaration types, such as [VertexPositionColor](#), [VertexPositionColorTexture](#), [VertexPositionNormalTexture](#), and [VertexPositionTexture](#). You may also use the [VertexElement](#) class to compose custom vertex types.

For a demonstration of low-level rendering of a 3D object from a vertex buffer, see [How To: Draw Points, Lines, and Other 3D Primitives](#).

Vertex buffers may contain either indexed or non-indexed vertex data.

If a vertex buffer is not indexed, all of the vertices are placed in the vertex buffer in the order they are to be rendered. Because 3D line lists or triangle lists often reference the same vertices multiple times, this can result in a large amount of redundant data.

Index buffers allow you to list each vertex only once in the vertex buffer. An index buffer is a list of indices into the vertex buffer, given in the order that you want the vertices to render.

To render a non-indexed vertex buffer, call the [GraphicsDevice.DrawPrimitives Method](#) or [GraphicsDevice.DrawUserPrimitives Generic Method](#). To render an indexed buffer, call the [GraphicsDevice.DrawIndexedPrimitives Method](#) or [GraphicsDevice.DrawUserIndexedPrimitives Method](#).

Textures

A texture resource is a structured collection of data designed to store texture data. The data in a texture resource is made up of one or more subresources that are organized into arrays and mipmap chains. Textures can be filtered by texture samplers as they are read by shaders. The type of texture influences how the texture is filtered.

You can apply textures by using the [Texture](#) property of the [BasicEffect](#) class, or choose to write your own effect methods to apply textures. For a demonstration of applying a texture from a user-created effect, see the example [How To: Create Custom](#)

[Texture Effects.](#)

Shaders and Effects

Shaders are programs that perform per-vertex or per-pixel processing of resources at run time. An effect is a combination of vertex and pixel shader code grouped together to encapsulate a particular rendering effect. Because shaders transform vertices and pixels at run time, they are used for lighting, materials, and textures. Unless you are using the [BasicEffect](#) class, you must first write shaders in either high-level shader language (HLSL) or assembly shader code (ASM), and then compile your shaders before your application can use the binary code.

For overviews of HLSL and the effect file format, see the [HLSL Shaders](#) and [Effects](#) DirectX Programming Guides on MSDN. Complete reference documentation for HLSL, shader ASM, and the effect file format is available in the [Direct3D API Reference](#).

For a simple demonstration of how to apply a vertex shader from an effect file, see [How To: Create and Apply Custom Effects](#).

See Also

Concepts

[Getting Started with 3D Games at XNA Creators Club Online](#)

[Shader Content Catalog at XNA Creators Club Online](#)

Viewports and Frustums

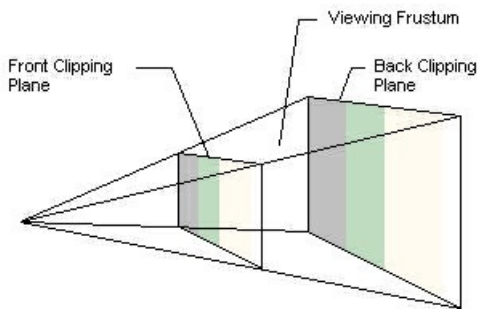
Provides overview of viewports and frustums as related to 3D graphics concepts.

Conceptually, a viewport is a two-dimensional (2D) rectangle onto which a 3D scene is projected. In the XNA Framework, the rectangle exists as x - and y -coordinates within a 2D surface used by XNA as a rendering target. Properly positioning graphical objects on this surface requires a mathematical transform (known as a projection transform). This transform converts 3D vertices into the coordinate system of the viewport. Additionally, a viewport can specify the range of depth values for the render-target surface. This range usually lies between 0.0 and 1.0.

- [The Viewing Frustum](#)
- [Viewport Rectangle](#)
- [Clearing a Viewport](#)

The Viewing Frustum

A viewing frustum is a 3D volume positioned relative to the viewport's camera. The shape of the volume affects how models are projected from camera space onto the screen. The most common type of projection, a perspective projection, is responsible for making objects near the camera appear bigger than objects in the distance. For perspective viewing, the viewing frustum can be visualized as a pyramid, with the camera positioned at the tip. This pyramid is intersected by a front and back clipping plane. The volume within the pyramid between the front and back clipping planes is referred to as the viewing frustum. Objects are visible only when they are in this volume.



Imagine that you are standing in a dark room and looking through a square window. This is a good representation of a viewing frustum. In this analogy, the near clipping plane is the window, and the back clipping plane is whatever finally interrupts your view - the skyscraper across the street, the mountains in the distance, or nothing at all. You can see everything inside the truncated pyramid that starts at the window and ends with whatever interrupts your view, and nothing else.

The viewing frustum is defined by a field of view (fov), and by the distances of the front and back clipping planes, which are specified in z -coordinates. You can create a perspective fov using the [CreatePerspectiveFieldOfView](#) method. For more information on setting up a frustum and perspective transformation, see [How To: Make a Third-Person Camera](#).

Viewport Rectangle

You define a viewport rectangle in XNA by using the [Viewport](#) structure. For more information about using this structure in your game application, see [Displays, Client Bounds, Viewports, and Back Buffers](#).

The [Viewport](#) structure contains four members - X , Y , $Width$, $Height$ - that define the area of the render-target surface in which a scene is rendered. These values correspond to the destination rectangle, or viewport rectangle.

The specified values for the X , Y , $Width$, and $Height$ members are screen coordinates relative to the upper-left corner of the render-target surface. The structure defines two additional members ($MinDepth$ and $MaxDepth$), indicating the near and far view planes.

Tip

The depth buffer has a limited resolution which is distributed between the near and far clip planes. If these planes are too far apart, two objects could have the same depth value even though they lie on different depth planes. In cases like this, the depth buffer is unable to distinguish between them because of its limited resolution.

To avoid this situation, set your near clipping plane as high as possible without causing near-plane clipping issues. For instance, avoid settings such as 0.000001, which would certainly cause problems.

The XNA Framework assumes that the viewport clipping volume ranges from -1.0 to 1.0 in X, and from 1.0 to -1.0 in Y. These were the settings used most often by applications in the past. You can specify the viewport aspect ratio, used for the projection transform, with a call to [CreatePerspectiveFieldOfView](#).

The XNA Framework uses the viewport location and dimensions to scale the vertices to fit a rendered scene into the appropriate location on the target surface. An internal matrix is applied to each vertex to scale vertices according to the viewport dimensions and desired depth range and translate them to the appropriate location on the render-target surface. This matrix also reverses the y-coordinate to reflect a screen origin at the top-left corner with y increasing downward. After this matrix is applied, vertices are still homogeneous. That is, they still exist as $[x,y,z,w]$ vertices—and they must be converted to non-homogeneous coordinates before being sent to the rasterizer.

Note

The viewport scaling matrix incorporates the `MinDepth` and `MaxDepth` members of the [Viewport](#) structure to scale vertices to fit the specified depth range.

Clearing a Viewport

Clearing the viewport resets the contents of the viewport rectangle on the render-target surface. It can also clear the depth and stencil buffer surfaces.

Use [Clear](#) to clear the various components of the viewport. Overloaded versions accept various parameters such as a set of rectangles, defining the areas on the surface to be cleared, or a [ClearOptions](#) enumeration that specifies which buffers or render target to clear.

Use the [Clear](#) method to clear stencil bits within a depth buffer. Set the options parameter to determine which render-target components are cleared. The color argument sets the color value of the render target after it has been cleared. The depth parameter clears the depth buffer to the specified depth: 0.0 is the closest distance, and 1.0 is the farthest. Finally, the stencil parameter resets the stencil bits to the specified value. Use integers ranging from 0 to $2^n - 1$, where n is the stencil buffer bit depth.

Dynamic Vertex Buffers in XNA

Describes several techniques for implementing dynamic vertex buffers in an XNA Framework game.

Geometry in a 3D game ultimately is defined as a set of vertices. Commonly, a game either needs to modify the position of one or more existing vertices or generate the vertices at run time. Because the graphics device can access these buffers at any time, take care when you modify existing vertex buffers and render the resulting geometry defined by these buffers. The following sections discuss several techniques you can implement in your game.

Tip

For index or vertex buffers that are frequently updated during game run time, use the [DynamicIndexBuffer](#) and [DynamicVertexBuffer](#) classes. They are optimized to render faster when updated frequently.

Use [VertexBuffer](#) and [IndexBuffer](#) when index or vertex buffer data is infrequently updated.

Updating a Set of Primitives Dynamically

You can use the [DrawUserPrimitives](#) method as the basis for rendering a set of primitives that are generated dynamically each frame. The [Primitives Sample](#), located on the XNA Creators Club Online Web site, implements an abstraction layer around this method similar to the [SpriteBatch](#) class.

The sample renders primitives by first calling `Begin`, adding the necessary vertices, using the `Add` method, and then calling `End`. This forces the buffer to be drawn to the current device. The `Flush` method calls [DrawUserPrimitives](#) method when `End` is called or when the buffer has no room for new vertices. If there is no room, the buffer is written to the device, it is reset, and the pending vertices are added.

For a complete code listing, download the [Primitives Sample](#) from the XNA Creators Club Online Web site.

Dynamically Rendering a Persistent Set of Primitives

The [Particle 3D Sample](#), located on the XNA Creators Club Online Web site, implements a dynamic vertex buffer that contains custom vertices with a limited lifespan. The application adds particles and removes them to a fixed length buffer. The custom shader of the sample renders the active subset of vertices dynamically.

Because particles have a limited lifespan, the sample code must manage the vertex buffer in real time. The `ParticleSystem` class handles all adding, updating, and deleting.

Generating Geometry Programmatically

Sometimes, geometry to be rendered either is not known at design-time or it changes based on events during run time.

To generate geometry

1. Create a vertex buffer to store the vertices of the geometry to be generated.
2. Create an index buffer that stores the indices used to render the geometry represented by the vertex buffer.
3. Once the vertex and index data have been generated, call [SetData](#) for the vertex buffer and [SetData](#) for the index buffer.
4. In the method that renders your frame (the `Draw` method for a default XNA game project), set the vertex stream and index data of the current graphics device.

For a specific example of this, see [DrawIndexedPrimitives](#).

5. Call [DrawIndexedPrimitives](#).

For a more advanced solution for this problem, download the [Generated Geometry Sample](#) from the XNA Creators Club Online Web site. This sample uses the [MeshBuilder](#) helper class and a custom processor to generate a terrain map from a bitmap loaded by the content manager. Specifically, examine the `Process` method, located in `TerrainProcessor.cs`. This method programmatically creates the terrain geometry based on input from the specified bitmap.

"How To" Articles

Describes step-by-step procedures for certain 3D graphics techniques.

In This Section

[How To: Check for Shader Model 2.0 Support](#)

Demonstrates how to find out whether a graphics card supports Shader Model 2.0.

[How To: Enable Antialiasing \(Multisampling\)](#)

Demonstrates how to use the [GraphicsDeviceManager](#) to enable antialiasing for your game. The [CheckDeviceMultiSampleType](#) method determines what kind of antialiasing your game computer supports.

[How To: Use BasicEffect](#)

Demonstrates how to create and initialize an instance of [BasicEffect](#), initialize a vertex buffer that can be rendered by [BasicEffect](#), apply the effect, and render the geometry.

[How To: Render a Model](#)

Demonstrates how to load and render a model using the XNA Framework Content Pipeline.

[How To: Create and Use a Custom Vertex](#)

Demonstrates how to create a custom vertex and use it to render a 3D primitive object.

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Demonstrates how to draw points, lines, and other 3D primitives.

[How To: Create and Apply Custom Effects](#)

Demonstrates how to create a simple effect to set the diffuse color and world, view, and projection transformation of a 3D object.

[How To: Draw a Model with a Custom Effect](#)

Demonstrates how to load a [Model](#) file and draw the [Model](#) using a custom [Effect](#) without modifying the Content Pipeline.

[How To: Create Custom Texture Effects](#)

Demonstrates how to create an effect to apply a texture to a 3D primitive object.

[How To: Draw a Textured Quad](#)

Demonstrates how to create and draw a simple quad using [DrawUserIndexedPrimitives](#).

[How To: Use Viewports for Split Screen Gaming](#)

Demonstrates how to use the [Viewport](#) property to display different scenes to different parts of the screen.

[How To: Create a SkySphere](#)

Demonstrates how to create an effect that will apply a skybox-style [TextureCube](#) ("cube map") to a sphere.

[How To: Draw a Shadow](#)

Demonstrates how to draw a shadow using [Matrix.CreateShadow](#) and the stencil buffer.

[How To: Use EffectParameters and EffectTechniques](#)

Demonstrates how to use [EffectParameters](#) objects to interact with effect variables, and to use [EffectTechniques](#) to access different techniques in an [Effect](#).

[How To: Create a Depth Texture](#)

Demonstrates how to create a texture that contains depth information for a scene using a customized [RenderTarget2D](#), [DepthStencilBuffer](#), and a simple [Effect](#).

[How To: Implement Shadow Mapping](#)

Demonstrates how to use [depth textures](#) (that is, shadow maps) to create dynamic shadows in a scene.

How To: Check for Shader Model 2.0 Support

Demonstrates how to find out whether a graphics card supports Shader Model 2.0.

Checking for Shader Model 2.0 Support

To check for Shader Model 2.0 support

1. Use the [GraphicsDevice.GraphicsDeviceCapabilities Property](#) to find the capabilities of the game's graphics device.
2. Compare [GraphicsDeviceCapabilities.MaxPixelShaderProfile](#) to the member of the [ShaderProfile](#) enumeration that represents pixel shader version ps_2_0.

In this example, if the graphics device does not support this shader model, the output window displays a message when you run the game in debug mode.

C#

```
// Check the graphics device used by the game
// for the necessary shader support.
GraphicsDeviceCapabilities caps =
    GraphicsDevice.GraphicsDeviceCapabilities;
if (caps.MaxPixelShaderProfile < ShaderProfile.PS_2_0)
{
    // This device does not support Shader Model 2.0.
    System.Diagnostics.Debug.WriteLine(
        "This adapter does not support Shader Model 2.0.");
}
```

How To: Enable Antialiasing (Multisampling)

Demonstrates how to use the [GraphicsDeviceManager](#) to enable antialiasing for your game. The [CheckDeviceMultiSampleType](#) method determines what kind of antialiasing your game computer supports.

Antialiasing is the process of smoothing the edges of triangles by averaging neighboring pixels in a render target. Because this process samples multiple pixels, it is also known as multisampling.

Note

Enabling antialiasing on the Xbox 360 will have a small performance impact. Enabling antialiasing on PCs may have a large impact on your game performance.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AntiAliasing_Sample.zip](#).

Using Antialiasing

To enable antialiasing in your game

- Set [PreferMultiSampling](#) to **true** in your [Game](#) class constructor.

This will default to [MultiSampleType.TwoSamples](#) on Xbox 360 (2x antialiasing). On a PC, the [GraphicsDeviceManager](#) will query for [MultiSampleType.NonMaskable](#), and select that, if possible. Otherwise, it will choose the best available antialiasing.

C#

```
graphics.PreferMultiSampling = true;
```

To customize antialiasing in your game

- In your [Game](#) class constructor, add an event handler to the [PreparingDeviceSettings](#) event on your [GraphicsDeviceManager](#).

If [PreferMultiSampling](#) is **true**, a [MultiSampleType](#) will be chosen automatically before the event is raised. Whether or not a [MultiSampleType](#) is chosen when the event is raised, you can use the event handler to query and choose your own [MultiSampleType](#).

C#

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
    graphics.PreferMultiSampling = true;
    graphics.PreparingDeviceSettings +=
        new EventHandler<PreparingDeviceSettingsEventArgs>(
            graphics_PreparingDeviceSettings);
}
```

- Create a method to use as an event handler for the [PreparingDeviceSettings](#) event.

You will be using the [PreparingDeviceSettingsEventArgs](#) instance to access the [PresentationParameters](#) and set your antialiasing preferences.

C#

```
void graphics_PreparingDeviceSettings(object sender,
    PreparingDeviceSettingsEventArgs e)
{
```

3. If your game is running on Xbox 360, set directly the [MultiSampleQuality](#) and [MultiSampleType](#) properties, and then return from your event handler.

The Xbox supports [MultiSampleType.TwoSamples](#) and [MultiSampleType.FourSamples](#), each with a [MultiSampleQuality](#) of 0 or 1. [MultiSampleType.FourSamples](#) is 4x antialiasing; [MultiSampleType.TwoSamples](#) is 2x antialiasing.

C#

```
// Xbox 360 and most PCs support FourSamples/0
// (4x) and TwoSamples/0 (2x) antialiasing.
PresentationParameters pp =
    e.GraphicsDeviceInformation.PresentationParameters;

#if XBOX
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType = MultiSampleType.FourSamples;
    return;
#else
```

4. If your game is running on a PC, use [CheckDeviceMultiSampleType](#) to query for antialiasing support.

Many PCs support 4x antialiasing, and most will support 2x antialiasing. Once you have determined that antialiasing is supported, set the [MultiSampleType](#) and [MultiSampleQuality](#) properties appropriately, and then return from the event handler.

C#

```
int quality = 0;
GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
SurfaceFormat format = adapter.CurrentDisplayMode.Format;
// Check for 4xAA
if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
    false, MultiSampleType.FourSamples, out quality))
{
    // even if a greater quality is returned, we only want quality 0
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType =
        MultiSampleType.FourSamples;
}
// Check for 2xAA
else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
    format, false, MultiSampleType.TwoSamples, out quality))
{
    // even if a greater quality is returned, we only want quality 0
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType =
        MultiSampleType.TwoSamples;
}
return;
#endif
}
```

See Also

Concepts

[3D Graphics Overview](#)

[What Is Antialiasing?](#)

Tasks

[How To: Display a Game in Full-Screen Mode](#)

Reference

[GraphicsDeviceManager](#)

[PreparingDeviceSettings](#)

[CheckDeviceMultiSampleType](#)

PresentationParameters

MultiSampleType

MultiSampleQuality

MultiSampleType

How To: Use BasicEffect

Demonstrates how to create and initialize an instance of [BasicEffect](#), initialize a vertex buffer that can be rendered by [BasicEffect](#), apply the effect, and render the geometry.

Note

The steps described here apply to effects created with the [BasicEffect](#) class. In contrast, the [Effect](#) class is provided for users who wish to write custom effects. For more information, see [How To: Create and Apply Custom Effects](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download UseBasicEffect_Sample.zip.](#)

Using BasicEffect

To use BasicEffect

Using the basic effect class requires a set of world, view, and projection matrices, a vertex buffer, a vertex declaration, and an instance of the [BasicEffect](#) class.

1. Declare these objects at the beginning of the game.

C#

```
Matrix worldMatrix;
Matrix viewMatrix;
Matrix projectionMatrix;
VertexPositionNormalTexture[] cubeVertices;
VertexDeclaration basicEffectVertexDeclaration;
VertexBuffer vertexBuffer;
BasicEffect basicEffect;
```

2. Initialize the world, view, and projection matrices.

In this example, you create a world matrix that rotates the geometry 22.5 degrees along the x and y axes. The view matrix is a look-at matrix with a camera position at (0, 0, 5), pointing at the origin. The projection matrix is a perspective projection matrix based on a 45-degree field of view, an aspect ratio equal to the client window, and a set of near and far planes.

C#

```
float tilt = MathHelper.ToRadians(22.5f); // 22.5 degree angle
// Use the world matrix to tilt the cube along x and y axes.
worldMatrix = Matrix.CreateRotationX(tilt) *
    Matrix.CreateRotationY(tilt);

viewMatrix = Matrix.CreateLookAt(new Vector3(0, 0, 5), Vector3.Zero,
    Vector3.Up);

projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.ToRadians(45), // 45 degree angle
    (float)GraphicsDevice.Viewport.Width /
    (float)GraphicsDevice.Viewport.Height,
    1.0f, 100.0f);
```

3. Initialize [BasicEffect](#) to the desired values.

C#

```
basicEffect = new BasicEffect(graphics.GraphicsDevice, null);
basicEffect.Alpha = 1.0f;
```



```

basicEffect.DiffuseColor = new Vector3(1.0f, 0.0f, 1.0f);
basicEffect.SpecularColor = new Vector3(0.25f, 0.25f, 0.25f);
basicEffect.SpecularPower = 5.0f;
basicEffect.AmbientLightColor = new Vector3(0.75f, 0.75f, 0.75f);

basicEffect.DirectionalLight0.Enabled = true;
basicEffect.DirectionalLight0.DiffuseColor = Vector3.One;
basicEffect.DirectionalLight0.Direction =
    Vector3.Normalize(new Vector3(1.0f, -1.0f, -1.0f));
basicEffect.DirectionalLight0.SpecularColor = Vector3.One;

basicEffect.DirectionalLight1.Enabled = true;
basicEffect.DirectionalLight1.DiffuseColor =
    new Vector3(0.5f, 0.5f, 0.5f);
basicEffect.DirectionalLight1.Direction =
    Vector3.Normalize(new Vector3(-1.0f, -1.0f, 1.0f));
basicEffect.DirectionalLight1.SpecularColor =
    new Vector3(0.5f, 0.5f, 0.5f);

basicEffect.LightingEnabled = true;

basicEffect.World = worldMatrix;
basicEffect.View = viewMatrix;
basicEffect.Projection = projectionMatrix;

```

4. Implement [BasicEffect](#) so it can use the vertex type that is appropriate for the options you will enable for the effect.

- If lighting is enabled, the vertex must have a normal type.
- If vertex colors are enabled, the vertex must have colors.
- If texturing is enabled, the vertex must have a texture coordinate.

The following code shows a cube created with vertices of type [VertexPositionNormalTexture](#).

C#

```

cubeVertices = new VertexPositionNormalTexture[36];

Vector3 topLeftFront = new Vector3(-1.0f, 1.0f, 1.0f);
Vector3 bottomLeftFront = new Vector3(-1.0f, -1.0f, 1.0f);
Vector3 topRightFront = new Vector3(1.0f, 1.0f, 1.0f);
Vector3 bottomRightFront = new Vector3(1.0f, -1.0f, 1.0f);
Vector3 topLeftBack = new Vector3(-1.0f, 1.0f, -1.0f);
Vector3 topRightBack = new Vector3(1.0f, 1.0f, -1.0f);
Vector3 bottomLeftBack = new Vector3(-1.0f, -1.0f, -1.0f);
Vector3 bottomRightBack = new Vector3(1.0f, -1.0f, -1.0f);

Vector2 textureTopLeft = new Vector2(0.0f, 0.0f);
Vector2 textureTopRight = new Vector2(1.0f, 0.0f);
Vector2 textureBottomLeft = new Vector2(0.0f, 1.0f);
Vector2 textureBottomRight = new Vector2(1.0f, 1.0f);

Vector3 frontNormal = new Vector3(0.0f, 0.0f, 1.0f);
Vector3 backNormal = new Vector3(0.0f, 0.0f, -1.0f);
Vector3 topNormal = new Vector3(0.0f, 1.0f, 0.0f);
Vector3 bottomNormal = new Vector3(0.0f, -1.0f, 0.0f);
Vector3 leftNormal = new Vector3(-1.0f, 0.0f, 0.0f);
Vector3 rightNormal = new Vector3(1.0f, 0.0f, 0.0f);

// Front face.

```

```

cubeVertices[0] =
    new VertexPositionNormalTexture(
        topLeftFront, frontNormal, textureTopLeft);
cubeVertices[1] =
    new VertexPositionNormalTexture(
        bottomLeftFront, frontNormal, textureBottomLeft);
cubeVertices[2] =
    new VertexPositionNormalTexture(
        topRightFront, frontNormal, textureTopRight);
cubeVertices[3] =
    new VertexPositionNormalTexture(
        bottomLeftFront, frontNormal, textureBottomLeft);
cubeVertices[4] =
    new VertexPositionNormalTexture(
        bottomRightFront, frontNormal, textureBottomRight);
cubeVertices[5] =
    new VertexPositionNormalTexture(
        topRightFront, frontNormal, textureTopRight);

// Back face.
cubeVertices[6] =
    new VertexPositionNormalTexture(
        topLeftBack, backNormal, textureTopRight);
cubeVertices[7] =
    new VertexPositionNormalTexture(
        topRightBack, backNormal, textureTopLeft);
cubeVertices[8] =
    new VertexPositionNormalTexture(
        bottomLeftBack, backNormal, textureBottomRight);
cubeVertices[9] =
    new VertexPositionNormalTexture(
        bottomLeftBack, backNormal, textureBottomRight);
cubeVertices[10] =
    new VertexPositionNormalTexture(
        topRightBack, backNormal, textureTopLeft);
cubeVertices[11] =
    new VertexPositionNormalTexture(
        bottomRightBack, backNormal, textureBottomLeft);

// Top face.
cubeVertices[12] =
    new VertexPositionNormalTexture(
        topLeftFront, topNormal, textureBottomLeft);
cubeVertices[13] =
    new VertexPositionNormalTexture(
        topRightBack, topNormal, textureTopRight);
cubeVertices[14] =
    new VertexPositionNormalTexture(
        topLeftBack, topNormal, textureTopLeft);
cubeVertices[15] =
    new VertexPositionNormalTexture(
        topLeftFront, topNormal, textureBottomLeft);
cubeVertices[16] =
    new VertexPositionNormalTexture(
        topRightFront, topNormal, textureBottomRight);
cubeVertices[17] =
    new VertexPositionNormalTexture(
        topRightBack, topNormal, textureTopRight);

```

```

// Bottom face.
cubeVertices[18] =
    new VertexPositionNormalTexture(
        bottomLeftFront, bottomNormal, textureTopLeft);
cubeVertices[19] =
    new VertexPositionNormalTexture(
        bottomLeftBack, bottomNormal, textureBottomLeft);
cubeVertices[20] =
    new VertexPositionNormalTexture(
        bottomRightBack, bottomNormal, textureBottomRight);
cubeVertices[21] =
    new VertexPositionNormalTexture(
        bottomLeftFront, bottomNormal, textureTopLeft);
cubeVertices[22] =
    new VertexPositionNormalTexture(
        bottomRightBack, bottomNormal, textureBottomRight);
cubeVertices[23] =
    new VertexPositionNormalTexture(
        bottomRightFront, bottomNormal, textureTopRight);

// Left face.
cubeVertices[24] =
    new VertexPositionNormalTexture(
        topLeftFront, leftNormal, textureTopRight);
cubeVertices[25] =
    new VertexPositionNormalTexture(
        bottomLeftBack, leftNormal, textureBottomLeft);
cubeVertices[26] =
    new VertexPositionNormalTexture(
        bottomLeftFront, leftNormal, textureBottomRight);
cubeVertices[27] =
    new VertexPositionNormalTexture(
        topLeftBack, leftNormal, textureTopLeft);
cubeVertices[28] =
    new VertexPositionNormalTexture(
        bottomLeftBack, leftNormal, textureBottomLeft);
cubeVertices[29] =
    new VertexPositionNormalTexture(
        topLeftFront, leftNormal, textureTopRight);

// Right face.
cubeVertices[30] =
    new VertexPositionNormalTexture(
        topRightFront, rightNormal, textureTopLeft);
cubeVertices[31] =
    new VertexPositionNormalTexture(
        bottomRightFront, rightNormal, textureBottomLeft);
cubeVertices[32] =
    new VertexPositionNormalTexture(
        bottomRightBack, rightNormal, textureBottomRight);
cubeVertices[33] =
    new VertexPositionNormalTexture(
        topRightBack, rightNormal, textureTopRight);
cubeVertices[34] =
    new VertexPositionNormalTexture(
        topRightFront, rightNormal, textureTopLeft);
cubeVertices[35] =
    new VertexPositionNormalTexture(
        bottomRightBack, rightNormal, textureBottomRight);

```

```
vertexBuffer = new VertexBuffer(  
    graphics.GraphicsDevice,  
    VertexPositionNormalTexture.SizeInBytes * cubeVertices.Length,  
    BufferUsage.None  
);  
  
vertexBuffer.SetData<VertexPositionNormalTexture>(cubeVertices);
```

5. Create a vertex declaration for the type [VertexPositionNormalTexture](#).

This vertex declaration is used in the render loop in the next step.

C#

```
basicEffectVertexDeclaration = new VertexDeclaration(  
    graphics.GraphicsDevice, VertexPositionNormalTexture.VertexElements);
```

6. Call the [Effect.Begin](#) to begin applying the [BasicEffect](#).
7. Draw the desired geometry between calls to [EffectPass.Begin](#) and [EffectPass.End](#).
8. Call [Effect.End](#) to stop application of the technique.

C#

```
graphics.GraphicsDevice.VertexDeclaration =  
    basicEffectVertexDeclaration;  
graphics.GraphicsDevice.Vertices[0].SetSource(vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);  
  
// This code would go between a device  
// BeginScene-EndScene block.  
basicEffect.Begin();  
foreach (EffectPass pass in basicEffect.CurrentTechnique.Passes)  
{  
    pass.Begin();  
  
    graphics.GraphicsDevice.DrawPrimitives(  
        PrimitiveType.TriangleList,  
        0,  
        12  
    );  
  
    pass.End();  
}  
basicEffect.End();
```

How To: Render a Model

Demonstrates how to load and render a model using the XNA Framework Content Pipeline. It is assumed that an existing Windows game project is loaded in XNA Game Studio. In this example, the project is called "CPModel."

This example has three main parts: importing and processing the model (design-time code), drawing the resultant managed object as a model with full lighting effect in the game (run-time code), and enabling movement of the model with the Xbox 360 game pad (run-time code).

Tip

This technique is implemented in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: Setting the Scene](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download CPModel_Sample.zip](#).

Adding and Rendering a Model

To add the model to the game

1. Right-click the **Content** project, click **Add**, and then click **Existing Item**.
2. From the **Files of type**: control, click **XNA Content Files**.
3. Navigate to the correct folder and select the model to be added.

For this example, use the wedge_player1.x file.

4. Click the small arrow to the right of the **Add** button, and then click **Add as Link**.

This creates a reference to the selected asset (and not a copy) in your Content project.

Tip

Making a reference to an existing file in your project is quite different from adding an existing item to your project. A file reference only stores the path to the file in question, not a copy of the contents. This is useful if the referenced file is dependent on additional external files and ensures that the solution always uses the latest version. This differs from adding a copy of the file to the project. In this case, it is possible to break external file dependencies and create extra work if the original content changes but the copy is not updated.

For these reasons, it is recommended that you place any media used by the solution in a subfolder of the solution, and that you add a reference to the media and not a copy of the media.

5. After you add the asset, open the **Properties** window and verify that the correct importer and processor are specified.

For this example, the **Content Importer** is X File-XNA Framework and the **Content Processor** is Model-XNA Framework. For ease of reference, the asset name value was manually changed to "ship" using the **Properties** window.

For more information on the **Properties** window of a game asset, see [Game Asset Properties](#).

6. Save the solution.

At this point, build the solution.

This completes the design-time portion of the example. The remaining parts render the model using the managed object (called "ship") and add some user control of the model. All code modifications for this part occur within the game1.cs file of the game project.

To render the model

1. From Solution Explorer, double-click the game1.cs file.
2. Modify the **Game1** class by adding the following code at the beginning of the declaration.

C#

```
private Model gameShip;
```

This member is used to hold the ship model.

3. Modify the **LoadGraphicsContent** method by adding the following code.

C#

```
gameShip = Content.Load<Model>("ship");
```

This code loads the model into the `gameShip` member (using [Load](#)).

4. Create a new **private** method (called **DrawModel**) in the **Game1** class by adding the following code before the existing [Draw](#) method.

C#

```
private void DrawModel(Model m)
{
    Matrix[] transforms = new Matrix[m.Bones.Count];
    float aspectRatio = graphics.GraphicsDevice.Viewport.Width /
        graphics.GraphicsDevice.Viewport.Height;
    m.CopyAbsoluteBoneTransformsTo(transforms);
    Matrix projection =
        Matrix.CreatePerspectiveFieldOfView(MathHelper.ToRadians(45.0f),
        aspectRatio, 1.0f, 10000.0f);
    Matrix view = Matrix.CreateLookAt(new Vector3(0.0f, 50.0f, Zoom),
        Vector3.Zero, Vector3.Up);

    foreach (ModelMesh mesh in m.Meshes)
    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.EnableDefaultLighting();

            effect.View = view;
            effect.Projection = projection;
            effect.World = gameWorldRotation *
                transforms[mesh.ParentBone.Index] *
                Matrix.CreateTranslation(Position);
        }
        mesh.Draw();
    }
}
```

This code sets up the lighting effects for each sub-mesh of the model. The `gameWorldRotation` and `Zoom` variables are used for player control. This functionality is added later.

Note

This render code is designed for only those models with a [BasicEffect](#). For custom effects, the inner `for-each` loop should be changed to use [Effect](#) instead of [BasicEffect](#). In addition, you must use [EffectParameter](#) objects to manually set the world, view, and projection matrices.

5. Modify the **Game1.Draw** method by replacing the following code // `TODO: Add your drawing code here` with the following code:

C#

```
DrawModel(gameShip);
```

This causes the effects initialization of the model before the entire model is rendered.

6. Save the solution.

At this point, the rendering code for the model is complete, but the user control code still needs implementation.

To move the model:

1. Modify the **Game1** class by adding the following code after the `gameShip` declaration.

C#

```
private Vector3 Position = Vector3.One;
private float Zoom = 2500;
private float RotationY = 0.0f;
private float RotationX = 0.0f;
private Matrix gameWorldRotation;
```

These members store the current position, zoom, and rotation values. In addition, the `gameWorldRotation` simplifies the `UpdateGamePad` code.

2. Add a private method (called **UpdateGamePad**) before the call to `Update`.

C#

```
private void UpdateGamePad()
{
    GamePadState state = GamePad.GetState(PlayerIndex.One);

    if (state.Buttons.A == ButtonState.Pressed)
    {
        Exit();
    }

    Position.X += state.ThumbSticks.Left.X * 10;
    Position.Y += state.ThumbSticks.Left.Y * 10;
    Zoom += state.ThumbSticks.Right.Y * 10;
    RotationY += state.ThumbSticks.Right.X;
    if (state.DPad.Up == ButtonState.Pressed)
    {
        RotationX += 1.0f;
    }
    else if (state.DPad.Down == ButtonState.Pressed)
    {
        RotationX -= 1.0f;
    }
    gameWorldRotation =
        Matrix.CreateRotationX(MathHelper.ToRadians(RotationX)) *
        Matrix.CreateRotationY(MathHelper.ToRadians(RotationY));
}
```

This code implements an exit method for the game (pressing the **A** button), and updates the position members with the current input of the game controller.

3. Modify the **Update** method by adding a call to `UpdateGamePad`, before the call to `Update`.

C#

```
UpdateGamePad();
```

This code updates the state of the position variables with the latest input.

4. Save the solution.

At this point, all development is complete. You can control the ship location with the game pad, and exit by pressing the **A** button.

See Also [Content Pipeline](#)

How To: Create and Use a Custom Vertex

Demonstrates how to create a custom vertex and use it to render a 3D primitive object. It extends the concepts discussed in [How To: Use BasicEffect](#) and [How To: Draw Points, Lines, and Other 3D Primitives](#).

A custom vertex can include any standard and custom data types. This enables you to provide functionality beyond the standard XNA Framework vertex declarations, declarations such as [VertexPositionColor](#).

This topic shows you the concept with a custom vertex (of type `VertexPositionColoredNormal`) with three properties: a [Vector3](#) position, a [Color](#) color, and a [Vector3](#) normal. With this information, you can render 3D primitives with basic color and lighting effects.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download UseCustomVertex_Sample.zip](#).

Declaring the Custom Vertex

The requirements for a custom vertex are:

- Properties, implemented using both standard and user-defined data types, containing the values for the custom vertex. For this example, three properties are declared representing the position, color, and normal of the vertex.
- A public method called **SizeInBytes**.
- An array of type [VertexElement](#) with each element representing a property of the custom vertex. For this example, the properties are represented by two [Vector3](#) types and a [Color](#) type.

In addition, you can implement some standard methods, such as constructors and public property accessors. For this example, a constructor and public methods for the three properties are implemented.

To declare the custom vertex

1. At the top of the source file declaring your game class, declare a structure containing the necessary components, based on the requirements listed previously.

This part of the structure declaration declares a custom vertex with properties for the position, color, and vertex normal.

C#

```

Vector3 vertexPosition;
Color vertexColor;
Vector3 vertexNormal;

public static int SizeInBytes { get { return 28; } }

//Declares the elements of the custom vertex.
//Each vertex stores information on the current
//position, color, and normal.
public static readonly VertexElement[] VertexElements =
    new VertexElement[] {
        new VertexElement(0, 0, VertexElementFormat.Vector3,
            VertexElementMethod.Default,
            VertexElementUsage.Position, 0),
        new VertexElement(0, sizeof(float) * 3,
            VertexElementFormat.Color, VertexElementMethod.Default,
            VertexElementUsage.Color, 0),
        new VertexElement(0, sizeof(float) * 3 + 4,
            VertexElementFormat.Vector3,
            VertexElementMethod.Default, VertexElementUsage.Normal,
            0) };

```

2. Declare any additional methods you think are necessary.

This part of the structure declaration declares a constructor and three public methods. There is a method for each custom vertex property.

C#

```
public VertexPositionColoredNormal(Vector3 pos, Color color,
    Vector3 normal)
{
    vertexPosition = pos;
    vertexColor = color;
    vertexNormal = normal;
}

//Public methods for accessing the components of the custom vertex.
public Vector3 Position { get { return vertexPosition; }
    set { vertexPosition = value; } }
public Color Color { get { return vertexColor; }
    set { vertexColor = value; } }
public Vector3 Normal { get { return vertexNormal; }
    set { vertexNormal = value; } }
```

The custom vertex structure declaration is now complete. Initializing the basic effect is next.

Initializing the Effect and Camera

The next step initializes an effect (of type [BasicEffect](#)) and the related camera. These are used for basic coloring and lighting effects, and for viewing the 3D object. Although this step is not strictly necessary for implementing a custom vertex, it is provided to show you a common usage scenario for this custom vertex, which is the rendering of a simple 3D object with color and light effects.

To initialize the basic effect and camera

1. Declare variables for the basic effect and various projection matrices.

C#

```
BasicEffect basicEffect;
Matrix worldMatrix;
Matrix viewMatrix;
Matrix projectionMatrix;
```

2. Initialize the matrices of the camera object.

In this topic, a translation matrix is created to place the primitive at the center of the viewport of the game. The camera is oriented to look at (0, 0, 0) from a short distance away.

C#

```
//Centers the triangle strip in the game's viewport.
worldMatrix = Matrix.CreateTranslation(-1.5f, -0.5f, 0.0f);

viewMatrix = Matrix.CreateLookAt(
    new Vector3(0.0f, 0.0f, 5.0f),
    Vector3.Zero,
    Vector3.Up
);

projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.ToRadians(45),
    (float)GraphicsDevice.Viewport.Width /
    (float)GraphicsDevice.Viewport.Height,
```

```
1.0f, 100.0f
);
```

3. Initialize the effect to display a colored 3D primitive with default lighting.

In this topic, vertex coloring and lighting are enabled. In addition, the default lighting scenario lights the primitive. This is the minimum needed to properly color and light the primitive.

C#

```
basicEffectVertexDeclaration =
    new VertexDeclaration(GraphicsDevice,
        VertexPositionColoredNormal.VertexElements);

//Enables some basic effect characteristics, such as vertex
//coloring and default lighting.
basicEffect = new BasicEffect(graphics.GraphicsDevice, null);
basicEffect.VertexColorEnabled = true;
basicEffect.LightingEnabled = true;
basicEffect.EnableDefaultLighting();

basicEffect.World = worldMatrix;
basicEffect.View = viewMatrix;
basicEffect.Projection = projectionMatrix;
```

This completes the initialization of the camera and basic effect. The next step sets up a basic triangle strip using the custom vertex declared earlier in this topic.

Initializing the 3D Primitive

In this step, you initialize a vertex array with custom vertices implemented by `VertexPositionColoredNormal`. In addition to an indices array, the vertex array later renders a colored triangle strip with default lighting effects.

To initialize the triangle strip

1. Declare an array (of type `VertexPositionColoredNormal`), and then initialize the various properties of the vertex elements.

In this topic, a triangle strip (from 0,0 to 3,0 and on the plane $z = 0$) is drawn.

C#

```
triangleStripVertices = new VertexPositionColoredNormal[8];

//Initialize the custom vertex values for the triangle strip.
for (int x = 0; x < 4; x++)
{
    for (int y = 0; y < 2; y++)
    {
        triangleStripVertices[(x * 2) + y] =
            new VertexPositionColoredNormal(
                new Vector3(x, y, 0.0f),
                Color.Red,
                new Vector3(0.0f, 0.0f, 1)
            );
    }
}
```

2. Create and initialize the indices buffer for the primitive.

Because a triangle strip is being used, the buffer is relatively small.

C#

```
triangleStripIndices = new short[8];
for (int i = 0; i < 8; i++)
{
    triangleStripIndices[i] = (short)i;
}
```

Initialization of the triangle strip is complete and the final step of rendering the triangle strip is next.

Drawing the 3D Primitive

In this step, a red triangle strip is rendered with some basic lighting effects. This code is commonly placed inside the [Draw](#) method of your game.

To draw the triangle strip

- Declare the proper vertex declaration for the graphics device, and call the [DrawUserIndexedPrimitives](#) method.

C#

```
GraphicsDevice.VertexDeclaration = basicEffectVertexDeclaration;

basicEffect.Begin();
foreach (EffectPass pass in basicEffect.CurrentTechnique.Passes)
{
    pass.Begin();

    GraphicsDevice.DrawUserIndexedPrimitives
        <VertexPositionColoredNormal>(
        PrimitiveType.TriangleStrip,
        triangleStripVertices,
        0, // vertex buffer offset
           // (to add to each element of the index buffer)
        8, // number of vertices to draw
        triangleStripIndices,
        0, // first index element to read
        6 // number of primitives to draw
    );

    pass.End();
}
basicEffect.End();
```

These simple lighting effects are the reason the `VertexPositionColoredNormal` custom vertex type was created. Because each vertex has a normal, along with a position and color, the basic effect determines where the triangle strip is highlighted and where any shadows lie. If you had instead used vertices of type `VertexPositionColor`, there would be no normals to determine any lighting effects.

How To: Draw Points, Lines, and Other 3D Primitives

Demonstrates how to draw points, lines, and other 3D primitives.

In the XNA Framework, a 3D primitive is a special type of 3D shape that describes how the graphics device interprets vertices stored in a vertex array or vertex stream. The following example shows you how to use the point, line, and triangle primitive types that are the basis for all low-level drawing calls in the XNA Framework.

Note

To render primitives, you need to create a basic effect and several transformation matrices. This topic follows the steps described in [How To: Use BasicEffect](#) to create an instance of [BasicEffect](#). This sample uses orthographic projection, but you can also use perspective projection to render primitives.

The vertices used in the sample are of type [VertexPositionColor Structure](#). Vertices of this type contain position and color information.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download DrawUserPrimitives_Sample.zip.](#)

The major steps for this example are listed below:

- [The Complete Sample](#)
- [Drawing Points](#)
- [Drawing Lines](#)
- [Drawing a Line Strip](#)
- [Drawing Triangles](#)
- [Drawing a Triangle Strip](#)

Drawing Points

To draw a point list

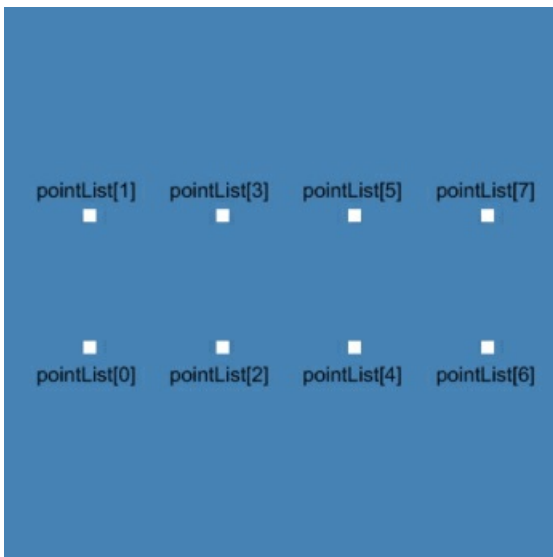
1. Create a list of vertices in 3D space that represent the points to draw.

The following code initializes eight points, and stores them in an array of type **VertexPositionColor**. This results in an array with the following vertex positions.

C#

```
pointList = new VertexPositionColor[points];

for (int x = 0; x < points / 2; x++)
{
    for (int y = 0; y < 2; y++)
    {
        pointList[(x * 2) + y] = new VertexPositionColor(
            new Vector3(x * 100, y * 100, 0), Color.White);
    }
}
```



These eight points form a triangle strip consisting of six triangles drawn along the plane $z = 0$, with the first point at $(0, 0, 0)$. The camera is positioned at $(0, 0, 1)$ looking at $(0, 0, 0)$. An orthogonal projection matrix is created with the upper-left point at $(0, 0)$ and the lower-right point at $(800, 600)$. In addition, a translation matrix shifts the point set to the center of the screen. The following code implements the positioning of the points, the camera, and the required transformations:

C#

```
viewMatrix = Matrix.CreateLookAt(  
    new Vector3(0.0f, 0.0f, 1.0f),  
    Vector3.Zero,  
    Vector3.Up  
);  
  
projectionMatrix = Matrix.CreateOrthographicOffCenter(  
    0,  
    (float)GraphicsDevice.Viewport.Width,  
    (float)GraphicsDevice.Viewport.Height,  
    0,  
    1.0f, 1000.0f);
```

2. Render the size of a point from a point list by setting the property [RenderState.PointSize](#).

The point size here is 10. You will see a 10-pixel square at each point in the list.

C#

```
GraphicsDevice.RenderState.PointSize = 10;
```

3. Render the points by calling [DrawUserPrimitives](#), which specifies [PrimitiveType.PointList](#) to determine how the data in the vertex array is interpreted.

C#

```
GraphicsDevice.DrawUserPrimitives<VertexPositionColor>(  
    PrimitiveType.PointList,  
    pointList,  
    0, // index of the first vertex to draw  
    8 // number of primitives  
);
```

Drawing Lines

The following example uses the sample vertex list created in step 1 of "To draw a point list."

To draw a line list

1. Create an index array that indexes into the vertex buffer.

This identifies a series of lines.

C#

```
// Initialize an array of indices of type short.
lineListIndices = new short[(points * 2) - 2];

// Populate the array with references to indices in the vertex buffer
for (int i = 0; i < points - 1; i++)
{
    lineListIndices[i * 2] = (short)(i);
    lineListIndices[(i * 2) + 1] = (short)(i + 1);
}
```

This is equivalent to setting `lineListIndices` to the following array, which consists of a series of lines between `pointList[0]` and `pointList[1]`, `pointList[1]` and `pointList[2]`, and so forth.

C#

```
lineListIndices = new short[14]{ 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7 };
```

2. Render the lines by calling `DrawUserIndexedPrimitives`, which specifies `PrimitiveType.LineList` to determine how to interpret the data in the vertex array.

C#

```
GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.LineList,
    pointList,
    0, // vertex buffer offset to add to each element of the index buffer
    8, // number of vertices in pointList
    lineListIndices, // the index buffer
    0, // first index element to read
    7 // number of primitives to draw
);
```

Drawing a Line Strip

The following example uses the same point list and renders the same output as "To draw a line list." However, it uses a line strip primitive type when it identifies the indices of the vertex array to draw. You store fewer indices when you use a line strip.

To draw a line strip

1. Create a list of indices to identify the order in which to draw the points in the specified point list.

You need only half the number of indices here as you used for the line list because the data consist of a series of connected lines.

C#

```
// Initialize an array of indices of type short.
lineStripIndices = new short[points];

// Populate the array with references to indices in the vertex buffer.
for (int i = 0; i < points; i++)
{
    lineStripIndices[i] = (short)(i);
}
```

This is equivalent to setting `lineStripIndices` to the following array, which consists of a series of *connected* lines between `pointList[0]`, `pointList[1]`, and `pointList[2]`, and so forth.

C#

```
lineStripIndices = new short[8]{ 0, 1, 2, 3, 4, 5, 6, 7 };
```

2. Render the line strip by calling `DrawUserIndexedPrimitives`, which specifies `PrimitiveType.LineStrip` to determine how to interpret the data in the vertex array.

Note that you use fewer vertices to render the same number of primitives rendered earlier by the line list.

Note

In the example code, the line strip is rendered by a series of red lines, instead of the white lines used for the previous line list. This is accomplished by changing the vertex color before you draw the line strip. After you complete the drawing, it reverts to the original color. In addition, vertex coloring was enabled for the basic effect (using `BasicEffect.VertexColorEnabled` Property).

The color change indicates a different primitive type was used to achieve the same result.

C#

```
for (int i = 0; i < pointList.Length; i++)
    pointList[i].Color = Color.Red;

GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.LineStrip,
    pointList,
    0, // vertex buffer offset to add to each element of the index buffer
    8, // number of vertices to draw
    lineStripIndices,
    0, // first index element to read
    7 // number of primitives to draw
);
for (int i = 0; i < pointList.Length; i++)
    pointList[i].Color = Color.White;
```

Drawing Triangles

Like a line list, a triangle list is a primitive type that indicates you need to interpret the vertices in the vertex buffer as a series of separately drawn triangles.

To draw a triangle list

1. Create an array to hold the list of indices that identify a series of triangles to draw from the specified point list.

C#

```
triangleListIndices = new short[(width - 1) * (height - 1) * 6];

for (int x = 0; x < width - 1; x++)
{
    for (int y = 0; y < height - 1; y++)
    {
        triangleListIndices[(x + y * (width - 1)) * 6] = (short)(2 * x);
        triangleListIndices[(x + y * (width - 1)) * 6 + 1] = (short)(2 * x + 1);
        triangleListIndices[(x + y * (width - 1)) * 6 + 2] = (short)(2 * x + 2);

        triangleListIndices[(x + y * (width - 1)) * 6 + 3] = (short)(2 * x + 2);
        triangleListIndices[(x + y * (width - 1)) * 6 + 4] = (short)(2 * x + 1);
        triangleListIndices[(x + y * (width - 1)) * 6 + 5] = (short)(2 * x + 3);
    }
}
```


This is equivalent to setting `triangleListIndices` to the following array, which consists of a series of triangles between `pointList[0]`, `pointList[1]`, and `pointList[2]`, and so forth.

C#

```
triangleListIndices = new short[18]{ 0, 1, 2, 2, 1, 3, 2, 3, 4, 4, 3, 5, 4, 5, 6, 6, 5, 7 };
```

2. Render the lines by calling `DrawUserIndexedPrimitives`

This specifies `PrimitiveType.TriangleList`, which determines how the data in the vertex array is interpreted.

C#

```
GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.TriangleList,
    pointList,
    0, // vertex buffer offset to add to each element of the index buffer
    8, // number of vertices to draw
    triangleListIndices,
    0, // first index element to read
    6 // number of primitives to draw
);
```

Drawing a Triangle Strip

A triangle strip is a set of triangles that share multiple vertices. This example shows you how to render an object that looks the same as the object rendered with a triangle list. However, fewer vertices are needed because the triangles share multiple vertices.

To draw a triangle strip

1. Create an array to hold the list of indices that identify a strip of triangles.

C#

```
// Initialize an array of indices of type short.
triangleStripIndices = new short[points];

// Populate the array with references to indices in the vertex buffer.
for (int i = 0; i < points; i++)
{
    triangleStripIndices[i] = (short)i;
}
```

This is equivalent to setting `triangleStripIndices` to the following array, which consists of a series of *connected* triangles between `pointList[0]`, `pointList[1]`, and `pointList[2]`, and so forth.

C#

```
triangleStripIndices = new short[8]{ 0, 1, 2, 3, 4, 5, 6, 7 };
```

2. Render the lines by calling `DrawUserIndexedPrimitives`.

This specifies `PrimitiveType.TriangleStrip` to determine how to interpret the data in the vertex array. Note that you use fewer vertices to render the same number of primitives rendered earlier by the triangle list.

Note

In the example code, the triangle strip is rendered by a series of red lines, instead of the white lines used for the previous triangle list. The color change indicates a different primitive type was used to achieve the same result.

C#

```
for (int i = 0; i < pointList.Length; i++)
```

```
    pointList[i].Color = Color.Red;

GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.TriangleStrip,
    pointList,
    0, // vertex buffer offset to add to each element of the index buffer
    8, // number of vertices to draw
    triangleStripIndices,
    0, // first index element to read
    6 // number of primitives to draw
);
for (int i = 0; i < pointList.Length; i++)
    pointList[i].Color = Color.White;
```

How To: Create and Apply Custom Effects

Demonstrates how to create a simple effect to set the diffuse color and world, view, and projection transformation of a 3D object.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ApplyAnEffect_Sample.zip.](#)

Applying an Effect

Note

The steps described here apply to effects created with the [Effect](#) class. In contrast, the [BasicEffect](#) class is provided for users who want basic lighting, materials, texture, and transformation functionality, but do not want to write custom effects. For more information, see [How To: Use BasicEffect](#).

To apply an effect

1. Compose the effect description using either shader assembly language (ASM) or high-level shader language (HLSL).

In this example, the effect file contains a vertex shader named **Transform** and a technique named **TransformTechnique**, which applies the vertex shader in a single pass.

```
uniform extern float4x4 WorldViewProj : WORLDVIEWPROJECTION;

struct VS_OUTPUT
{
    float4 position : POSITION;
    float4 color : COLOR0;
};

VS_OUTPUT Transform(
    float4 Pos : POSITION,
    float4 Color : COLOR0 )
{
    VS_OUTPUT Out = (VS_OUTPUT)0;

    Out.position = mul(Pos, WorldViewProj);
    Out.color = Color;

    return Out;
}

float4 PixelShader( VS_OUTPUT vsout ) : COLOR
{
    return vsout.color;
}

technique TransformTechnique
{
    pass P0
    {
        vertexShader = compile vs_2_0 Transform();
        pixelShader = compile ps_1_1 PixelShader();
    }
}
```

2. Add the effect file to the Content directory of your XNA Framework game.

3. Right-click your Content node, point to **Add**, and then click **Existing Item**.
4. In the **Files of type:** drop-down list, click **All Files**.
5. Navigate to your .fx file, and click **Add**.

Tip

To create a new .fx file, point to **Add**, click **New Item**, and then select the **Text File** template. For the name of the file, use the .fx file extension.

6. Create a new [Effect](#) by using the [ContentManager.Load<Effect>](#) method to load the asset.

C#

```
effect = Content.Load<Effect>("ReallySimpleEffect");
```

In the effect file, there is a parameter named *WorldViewProj*, which is a matrix used by the vertex shader. The following code looks up the *WorldViewProj* parameter in [Effect.Parameters](#), and calls [EffectParameter.SetValue](#) to initialize the value of the transform matrix.

C#

```
effect.Parameters["WorldViewProj"].SetValue(worldViewProjection);
```

7. Set [Effect.CurrentTechnique](#) to the technique from the effect file you wish to apply.

In this case, use [Effect.Techniques](#) to look up the technique named **TransformTechnique** from the effect file.

C#

```
effect.CurrentTechnique = effect.Techniques["TransformTechnique"];
```

8. Call the [Effect.Begin](#) to begin applying the technique.
9. For each pass in the technique, draw the desired geometry between calls to [EffectPass.Begin](#) and [EffectPass.End](#).
10. Call [Effect.End](#) to stop applying the technique.

C#

```
// The effect is a compiled effect created and compiled elsewhere
// in the application.

effect.Begin();
foreach (EffectPass pass in effect.CurrentTechnique.Passes)
{
    pass.Begin();

    GraphicsDevice.DrawIndexedPrimitives(
        PrimitiveType.TriangleList,
        0,
        0,
        cubeVertices.Length,
        0,
        12);

    pass.End();
}
effect.End();
```

See Also

Concepts

[Shader Content Catalog at XNA Creators Club Online](#)

How To: Draw a Model with a Custom Effect

Demonstrates how to load a [Model](#) file and draw the [Model](#) using a custom [Effect](#) without modifying the Content Pipeline.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ShadowMapping_Sample.zip.](#)

Drawing a Model With a Custom Effect

When you load a [Model](#), the model is set by default to use the [BasicEffect](#). You can change this by customizing the content pipeline, or you can apply a new [Effect](#) to the [Model](#) when you load the [Model](#).

To draw a model with a custom effect

1. In your game's [LoadContent](#) method, load your [Model](#), typically using the [ContentManager](#).

C#

```
terrain = Content.Load<Model>("terrain");
```

2. In [LoadContent](#), load your [Effect](#), typically using the [ContentManager](#).

C#

```
MyEffect = content.Load<Effect>("CustomEffect");
```

3. Iterate through each [ModelMeshPart](#) in your model, and assign your [Effect](#) to the [Effect](#) property of the [ModelMeshPart](#).

C#

```
public static void RemapModel(Model model, Effect effect)
{
    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (ModelMeshPart part in mesh.MeshParts)
        {
            part.Effect = effect;
        }
    }
}
```

4. Draw the [Model](#), using the steps outlined in [How To: Render a Model](#) with one exception: instead of using [BasicEffect](#), use the [Effect](#) attached to the model.

C#

```
foreach (ModelMesh mesh in terrain.Meshes)
{
    foreach (Effect effect in mesh.Effects)
    {
        mesh.Draw();
    }
}
```

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Render a Model](#)

Reference

[Model](#)

[ModelMeshPart](#)

[Effect](#)

[Draw](#)

[Effect](#)

How To: Create Custom Texture Effects

Demonstrates how to create an effect to apply a texture to a 3D primitive object. This topic builds on the demonstrations [How To: Draw Points, Lines, and Other 3D Primitives](#) and [How To: Create and Apply Custom Effects](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ApplyTextures_Sample.zip.](#)

The major steps for this example are listed below:

- [Composing the Texture](#)
- [Associating Vertices with Texture Coordinates](#)
- [Initializing the Texture](#)
- [Initializing the Effect](#)
- [Applying the Effect Technique](#)

Composing the Texture

To compose the texture effect

- Compose the effect description using either shader assembly language (ASM) or high-level shader language (HLSL).

In this example, the effect file contains a vertex shader to transform the object and a pixel shader to apply the texture to the object.

Note

In this effect file, even though the vertex shader does not manipulate the texture coordinate, it returns the texture coordinate so that it will be available to the pixel shader.

```
uniform extern float4x4 WorldViewProj : WORLDVIEWPROJECTION;
uniform extern texture UserTexture;

struct VS_OUTPUT
{
    float4 position : POSITION;
    float4 textureCoordinate : TEXCOORD0;
};

sampler textureSampler = sampler_state
{
    Texture = <UserTexture>;
    mipfilter = LINEAR;
};

VS_OUTPUT Transform(
    float4 Position : POSITION,
    float4 TextureCoordinate : TEXCOORD0 )
{
    VS_OUTPUT Out = (VS_OUTPUT)0;

    Out.position = mul(Position, WorldViewProj);
    Out.textureCoordinate = TextureCoordinate;

    return Out;
}

float4 ApplyTexture(VS_OUTPUT vsout) : COLOR
```

```

{
    return tex2D(textureSampler, vsout.textureCoordinate).rgba;
}

technique TransformAndTexture
{
    pass P0
    {
        vertexShader = compile vs_2_0 Transform();
        pixelShader  = compile ps_2_0 ApplyTexture();
    }
}

```

Associating Vertices with Texture Coordinates

To associate texture coordinates with vertices

1. Create a vertex declaration of type [VertexPositionTexture](#).

C#

```

cubeVertexDeclaration = new VertexDeclaration(
    GraphicsDevice, VertexPositionTexture.VertexElements);

```

2. Initialize the points to be used to draw each side of the cube.

C#

```

Vector3 topLeftFront = new Vector3(-1.0f, 1.0f, 1.0f);
Vector3 bottomLeftFront = new Vector3(-1.0f, -1.0f, 1.0f);
Vector3 topRightFront = new Vector3(1.0f, 1.0f, 1.0f);
Vector3 bottomRightFront = new Vector3(1.0f, -1.0f, 1.0f);
Vector3 topLeftBack = new Vector3(-1.0f, 1.0f, -1.0f);
Vector3 topRightBack = new Vector3(1.0f, 1.0f, -1.0f);
Vector3 bottomLeftBack = new Vector3(-1.0f, -1.0f, -1.0f);
Vector3 bottomRightBack = new Vector3(1.0f, -1.0f, -1.0f);

```

3. Initialize the texture coordinates.

C#

```

Vector2 textureTopLeft = new Vector2(0.0f, 0.0f);
Vector2 textureTopRight = new Vector2(1.0f, 0.0f);
Vector2 textureBottomLeft = new Vector2(0.0f, 1.0f);
Vector2 textureBottomRight = new Vector2(1.0f, 1.0f);

```

4. Declare an array to hold the list of vertices.

This array will be used to assign data to the vertex buffer.

C#

```

cubeVertices = new VertexPositionTexture[36];

```

5. Assign the position and texture coordinate data to each element of the vertex array.

C#

```

// Vertices for the front of the cube.
cubeVertices[0] =
    new VertexPositionTexture(
        topLeftFront, textureTopLeft); // 0
cubeVertices[1] =

```



```

    new VertexPositionTexture(
        bottomLeftFront, textureBottomLeft); // 1
cubeVertices[2] =
    new VertexPositionTexture(
        topRightFront, textureTopRight); // 2
cubeVertices[3] =
    new VertexPositionTexture(
        bottomRightFront, textureBottomRight); // 3

// Vertices for the back of the cube.
cubeVertices[4] =
    new VertexPositionTexture(
        topLeftBack, textureTopRight); // 4
cubeVertices[5] =
    new VertexPositionTexture(
        topRightBack, textureTopLeft); // 5
cubeVertices[6] =
    new VertexPositionTexture(
        bottomLeftBack, textureBottomRight); //6
cubeVertices[7] =
    new VertexPositionTexture(
        bottomRightBack, textureBottomLeft); // 7

// Vertices for the top of the cube.
cubeVertices[8] =
    new VertexPositionTexture(
        topLeftFront, textureBottomLeft); // 8
cubeVertices[9] =
    new VertexPositionTexture(
        topRightBack, textureTopRight); // 9
cubeVertices[10] =
    new VertexPositionTexture(
        topLeftBack, textureTopLeft); // 10
cubeVertices[11] =
    new VertexPositionTexture(
        topRightFront, textureBottomRight); // 11

// Vertices for the bottom of the cube.
cubeVertices[12] =
    new VertexPositionTexture(
        bottomLeftFront, textureTopLeft); // 12
cubeVertices[13] =
    new VertexPositionTexture(
        bottomLeftBack, textureBottomLeft); // 13
cubeVertices[14] =
    new VertexPositionTexture(
        bottomRightBack, textureBottomRight); // 14
cubeVertices[15] =
    new VertexPositionTexture(
        bottomRightFront, textureTopRight); // 15

// Vertices for the left side of the cube.
cubeVertices[16] =
    new VertexPositionTexture(
        topLeftFront, textureTopRight); // 16
cubeVertices[17] =
    new VertexPositionTexture(
        bottomLeftFront, textureBottomRight); // 17
cubeVertices[18] =

```

```
    new VertexPositionTexture(
        topRightFront, textureTopLeft); // 18
cubeVertices[19] =
    new VertexPositionTexture(
        bottomRightFront, textureBottomLeft); // 19
```

6. Create a vertex buffer to hold the vertex data.

C#

```
vertexBuffer = new VertexBuffer(GraphicsDevice,
    VertexPositionTexture.SizeInBytes * cubeVertices.Length,
    BufferUsage.None
);
```

7. Add the data to the vertex buffer.

C#

```
vertexBuffer.SetData<VertexPositionTexture>(cubeVertices);
```

8. Create indices to index into the **cubeVertices** array.

C#

```
cubeIndices = new short[] {
    0, 1, 2, // front face
    1, 3, 2,
    4, 5, 6, // back face
    6, 5, 7,
    8, 9, 10, // top face
    8, 11, 9,
    12, 13, 14, // bottom face
    12, 14, 15,
    16, 13, 17, // left face
    10, 13, 16,
    18, 19, 14, // right face
    9, 18, 14 };
```

9. Create an index buffer to hold the index array data.

C#

```
indexBuffer = new IndexBuffer(GraphicsDevice,
    sizeof(short) * cubeIndices.Length,
    BufferUsage.None,
    IndexElementSize.SixteenBits
);
```

10. Add the data to the index buffer.

C#

```
indexBuffer.SetData<short>(cubeIndices);
```

Initializing the Texture

Now you will want to add texture to the Content directory of your game project.

To initialize the texture

1. From Solution Explorer, right-click the Content node of your game project, click **Add**, and then click **Existing Item**.

2. In the **Files of type:** drop-down list, select **All Files**.
3. Navigate to your texture file and click **Add**.

In this example, a texture named xna.jpg is added and assigned a corresponding asset name of **xna**. For more information about game asset properties, see [Game Asset Properties](#).

4. Create a new [Texture2D](#) object using the [ContentManager.Load<Texture2D>](#) method to load the asset.

C#

```
Texture2D texture = Content.Load<Texture2D>("xna");
```

Initializing the Effect

This effect has two parameters, a transformation matrix named **WorldViewProj** to use in the vertex shader, and the texture named **UserTexture** to use in the texture sampler to be applied to the pixel shader. You need to set these parameters to the corresponding objects in the application: the [Matrix](#) object named **worldViewProjection**, and the [Texture2D](#) object named **texture**.

C#

```
effect.Parameters["WorldViewProj"].SetValue(worldViewProjection);  
effect.Parameters["UserTexture"].SetValue(texture);
```

Applying the Effect Technique

There is only one technique available in the effect file, so in this example [CurrentTechnique](#) is set to **TransformAndTexture**, which is the name of the technique in the effect file.

C#

```
effect.CurrentTechnique = effect.Techniques["TransformAndTexture"];
```

To select and apply a texture technique

1. Set the device vertex stream, indices, and vertex declaration to correspond to the vertices for which the texture is to be applied.

C#

```
GraphicsDevice.VertexDeclaration = cubeVertexDeclaration;  
GraphicsDevice.Indices = indexBuffer;  
GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer,  
    0,  
    VertexPositionTexture.SizeInBytes);
```

2. Apply the technique by placing the drawing calls between [EffectPass.Begin](#) and [EffectPass.End](#) inside an [Effect.Begin ... Effect.End](#) block.

C#

```
// This code would go between a device BeginScene-EndScene block.  
effect.Begin();  
foreach (EffectPass pass in effect.CurrentTechnique.Passes)  
{  
    pass.Begin();  
  
    GraphicsDevice.DrawIndexedPrimitives(  
        PrimitiveType.TriangleList,  
        0,  
        0,  
        cubeVertices.Length,  
        0,
```

12

);

pass.End();

}

effect.End();

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Use BasicEffect](#)

[How To: Use EffectParameters and EffectTechniques](#)

How To: Draw a Textured Quad

Demonstrates how to create and draw a simple quad using `DrawUserIndexedPrimitives`.

A quad is two triangles that form a rectangle or square. This sample introduces the **Quad** class. This class constructs a quad with a list of vertices and indices suitable for drawing with `DrawUserIndexedPrimitives`. This sample also demonstrates how to use `BasicEffect` to apply a texture to the primitive.

Note

To render the quad, this sample uses the `BasicEffect` class. For a discussion of `BasicEffect` see [How To: Use BasicEffect](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download TexturedQuad_Sample.zip](#).

The Textured Quad

To Create a Quad

1. Determine the location of each of the four vertices in the quad.

The Quad constructor calculates the four corners given the origin, width, height, and facing information supplied by the caller.

C#

```
public Quad( Vector3 origin, Vector3 normal, Vector3 up,
            float width, float height )
{
    Vertices = new VertexPositionNormalTexture[4];
    Indexes = new int[6];
    Origin = origin;
    Normal = normal;
    Up = up;

    // Calculate the quad corners
    Left = Vector3.Cross( normal, Up );
    Vector3 uppercenter = (Up * height / 2) + origin;
    UpperLeft = uppercenter + (Left * width / 2);
    UpperRight = uppercenter - (Left * width / 2);
    LowerLeft = UpperLeft - (Up * height);
    LowerRight = UpperRight - (Up * height);

    FillVertices();
}
```

2. Determine the UV coordinates for the texture you want to apply to the quad.

The Quad constructor calls `FillVertices`, which assigns UV coordinates ranging from (0,0) to (1,1). This will make the entire texture appear on the quad.

C#

```
private void FillVertices()
{
    // Fill in texture coordinates to display full texture
    // on quad
    Vector2 textureUpperLeft = new Vector2( 0.0f, 0.0f );
    Vector2 textureUpperRight = new Vector2( 1.0f, 0.0f );
    Vector2 textureLowerLeft = new Vector2( 0.0f, 1.0f );
}
```

```
Vector2 textureLowerRight = new Vector2( 1.0f, 1.0f );
```

- Determine the normals for your vertices.

FillVertices copies the normal vector you provide to each vertex.

C#

```
// Provide a normal for each vertex
for (int i = 0; i < Vertices.Length; i++)
{
    Vertices[i].Normal = Normal;
}
```

- Copy the position and texture coordinate data to your vertex array.

C#

```
// Set the position and texture coordinate for each
// vertex
Vertices[0].Position = LowerLeft;
Vertices[0].TextureCoordinate = textureLowerLeft;
Vertices[1].Position = UpperLeft;
Vertices[1].TextureCoordinate = textureUpperLeft;
Vertices[2].Position = LowerRight;
Vertices[2].TextureCoordinate = textureLowerRight;
Vertices[3].Position = UpperRight;
Vertices[3].TextureCoordinate = textureUpperRight;
```

- Fill out the index buffer to determine what order your vertices are drawn by the graphics hardware.

Drawing two triangles requires four vertices, but six index entries if you are using [PrimitiveType.TriangleList](#). The indices are specified in clockwise order. Because XNA is a right-handed system, triangles drawn in counter-clockwise order are assumed to be facing away from the camera, and are automatically culled by default.

C#

```
// Set the index buffer for each vertex, using
// clockwise winding
Indexes[0] = 0;
Indexes[1] = 1;
Indexes[2] = 2;
Indexes[3] = 2;
Indexes[4] = 1;
Indexes[5] = 3;
}
```

To Draw a Quad

- In your game's [Initialize](#) method, create a new **Quad** object specifying the location, the normal (the direction the Quad faces), the Up vector, and the width and height of the quad.

You can also create a View and Projection matrix to use when rendering.

C#

```
Quad quad;
VertexDeclaration quadVertexDecl;
Matrix View, Projection;
protected override void Initialize()
{
    quad = new Quad(Vector3.Zero, Vector3.Backward, Vector3.Up, 1, 1);
    View = Matrix.CreateLookAt(new Vector3(0, 0, 2), Vector3.Zero,
```

```

        Vector3.Up);
    Projection = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.PiOver4, 4.0f / 3.0f, 1, 500);

    base.Initialize();
}

```

- In your game's `LoadContent` method, load the texture you want to apply to the quad using the `ContentManager`.
- Create and initialize a new `BasicEffect` object.
- Set `TextureEnabled` to `true`, and assign the texture to draw to the `Texture` property.

C#

```

Texture2D texture;
BasicEffect quadEffect;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    texture = Content.Load<Texture2D>("Glass");
    quadEffect = new BasicEffect(graphics.GraphicsDevice, null);
    quadEffect.EnableDefaultLighting();

    quadEffect.World = Matrix.Identity;
    quadEffect.View = View;
    quadEffect.Projection = Projection;
    quadEffect.TextureEnabled = true;
    quadEffect.Texture = texture;
}

```

- In `LoadContent`, create a `VertexDeclaration` for the vertex type used to define the quad, in this case, `VertexPositionNormalTexture`.

C#

```

quadVertexDecl = new VertexDeclaration(graphics.GraphicsDevice,
    VertexPositionNormalTexture.VertexElements);
}

```

- In your game's `Draw` method, set the `VertexDeclaration` on the `GraphicsDevice` to the `VertexDeclaration` object you created in `LoadContent`.

C#

```

protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);
    GraphicsDevice.VertexDeclaration = quadVertexDecl;
}

```

- Call `Begin` on the effect.
- Use `DrawUserIndexedPrimitives` in each `EffectPass` to draw the quad.
- Use the `Vertices` and `Indices` properties on the `Quad` structure to supply the primitive data, and specify two triangles to draw.
- Call `End` on the effect.

C#

```

quadEffect.Begin();
foreach (EffectPass pass in quadEffect.CurrentTechnique.Passes)

```

```
{
    pass.Begin();

    GraphicsDevice.DrawUserIndexedPrimitives
        <VertexPositionNormalTexture>(
        PrimitiveType.TriangleList,
        quad.Vertices, 0, 4,
        quad.Indexes, 0, 2);

    pass.End();
}
quadEffect.End();

base.Draw(gameTime);
}
```

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Use BasicEffect](#)

Reference

[BasicEffect](#)

[VertexDeclaration](#)

[DrawUserIndexedPrimitives](#)

How To: Use Viewports for Split Screen Gaming

Demonstrates how to use the [Viewport](#) property to display different scenes to different parts of the screen. This sample assumes there are two camera classes in use, **Camera1** and **Camera2**.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download SplitScreen_Sample.zip.](#)

Creating a Split Screen

To create a split screen

1. In your [LoadContent](#) method, create two new [Viewport](#) objects to define the two new "split" regions of the screen.

In this example, the screen is split in half vertically.

C#

```
Viewport defaultViewport;
Viewport leftViewport;
Viewport rightViewport;
Matrix projectionMatrix;
Matrix halfprojectionMatrix;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    defaultViewport = GraphicsDevice.Viewport;
    leftViewport = defaultViewport;
    rightViewport = defaultViewport;
    leftViewport.Width = leftViewport.Width / 2;
    rightViewport.Width = rightViewport.Width / 2;
    rightViewport.X = leftViewport.Width + 1;
}
```

2. Create a projection matrix to fit each new viewport.

In this case, because the screen is split in half, only one new projection matrix is necessary. It has the same settings as the 4:3 full screen projection matrix, except the aspect ratio is now 2:3.

C#

```
projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.PiOver4, 4.0f / 3.0f, 1.0f, 10000f);
halfprojectionMatrix = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.PiOver4, 2.0f / 3.0f, 1.0f, 10000f);
}
```

3. In your [Draw](#) method, assign one of the viewports to draw as the [GraphicsDevice Viewport](#).
4. Draw your scene as normal, using the camera (or view matrix) associated with this perspective along with the proper projection matrix.

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Viewport = defaultViewport;
    GraphicsDevice.Clear(Color.CornflowerBlue);
}
```

```
GraphicsDevice.Viewport = leftViewport;  
DrawScene(gameTime, Camera1.ViewMatrix, halfprojectionMatrix);
```

5. After drawing the first scene, assign the other viewport to the [Viewport](#) property.
6. Draw your scene again with the associated camera or view matrix, and the proper projection matrix.

C#

```
GraphicsDevice.Viewport = rightViewport;  
DrawScene(gameTime, Camera2.ViewMatrix, halfprojectionMatrix);  
  
base.Draw(gameTime);  
  
}
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[GraphicsDevice.Viewport](#)

[Viewport Structure](#)

How To: Create a SkySphere

Demonstrates how to create an effect that will apply a skybox-style TextureCube ("cube map") to a sphere.

This example assumes you have a Camera object for handling the position and view matrix of the 3D camera, along with a simple [Model](#) to display. The SkySphere effect also requires a sphere model and a skybox cube map.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download SkySphere_Sample.zip.](#)

The Skysphere Effect

The vertex shader for the SkySphere effect takes the position of each vertex and extracts the rotation of the camera from the view matrix. It then applies the projection matrix to that for the final vertex position. For calculating the pixel values between the vertices, the vertex shader passes the unmodified position.

C#

```
uniform extern float4x4 ViewMatrix;
uniform extern float4x4 ProjectionMatrix;

void SkyboxVertexShader( float3 pos : POSITION0,
                        out float4 SkyPos : POSITION0,
                        out float3 SkyCoord : TEXCOORD0 )
{
    // Calculate rotation. Using a float3 result, so translation is ignored
    float3 rotatedPosition = mul(pos, ViewMatrix);
    // Calculate projection, moving all vertices to the far clip plane
    // (w and z both 1.0)
    SkyPos = mul(float4(rotatedPosition, 1), ProjectionMatrix).xyww;

    SkyCoord = pos;
};
```

The pixel shader for the SkySphere effect uses the texCUBE function to choose a pixel on the cube map.

To Create the SkySphere Effect

1. Imagine the sphere model surrounding the camera viewpoint, and the cube map surrounding the sphere model (as a cube).
2. Draw a line from the camera position to a point on the sphere model, and continue that line until it hits the cube map.

This is the pixel returned by texCUBE. This has the effect of transforming the cube into a different shape, which makes the sky look more realistic (the cube often has obvious corners). You do not have to use a sphere for this effect. A flattened or oblong sphere may look better for some applications.

C#

```
uniform extern texture SkyboxTexture;
sampler SkyboxS = sampler_state
{
    Texture = <SkyboxTexture>;
    MinFilter = LINEAR;
    MagFilter = LINEAR;
    MipFilter = LINEAR;
    AddressU = CLAMP;
    AddressV = CLAMP;
};
float4 SkyboxPixelShader( float3 SkyCoord : TEXCOORD0 ) : COLOR
{
```

```

    // grab the pixel color value from the skybox cube map
    return texCUBE(SkyboxS, SkyCoord);
};

```

The technique for the SkySphere effect has one pass. Because you want all the triangles on the sphere to render, the technique has to disable culling. Since the camera is inside the sphere, normal culling would result in no vertices to shade. Depth writes are also disabled because we want the environment to render behind everything else, without appearing to clip any objects.

C#

```

technique SkyboxTechnique
{
    pass P0
    {
        vertexShader = compile vs_2_0 SkyboxVertexShader();
        pixelShader = compile ps_2_0 SkyboxPixelShader();

        // We're drawing the inside of a model
        CullMode = None;
        // We don't want it to obscure objects with a Z < 1
        ZWriteEnable = false;
    }
}

```

To Apply the SkySphere Effect

1. In your `Game.LoadContent`, load the content that you normally draw in your scene.

C#

```

Vector3 ModelPosition;
float ModelRotation = 0.0f;
Model Model;
Model SkySphere;
Effect SkySphereEffect;
Matrix projectionMatrix;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    Model = Content.Load<Model>("redtorus");
    ModelPosition = Vector3.Zero;
    projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.PiOver4, 4.0f / 3.0f, 1.0f, 10000f);
}

```

2. Load the SkySphere effect from the Content Manager.
3. Load a cube map into a `TextureCube` object.

A cube map is a texture with six sides that form a cube. For a skybox (or sphere), the cube map is a picture of the far distance in the scene.

4. Set the parameters the SkySphere requires, including the `TextureCube` to draw.

C#

```

// Load the effect, the texture it uses, and
// the model used for drawing it
SkySphereEffect = Content.Load<Effect>("SkySphere");
TextureCube SkyboxTexture =
    Content.Load<TextureCube>("uffizi_cross");

```

```

SkySphere = Content.Load<Model>("SphereHighPoly");

// Set the parameters of the effect
SkySphereEffect.Parameters["ViewMatrix"].SetValue(
    myCamera.ViewMatrix);
SkySphereEffect.Parameters["ProjectionMatrix"].SetValue(
    projectionMatrix);
SkySphereEffect.Parameters["SkyboxTexture"].SetValue(
    SkyboxTexture);

```

5. Since the [Effect](#) and the [Model](#) are loaded separately, apply the SkySphere effect to each [Effect](#) property on the [ModelMeshPart](#) of the SkySphere model.

C#

```

// Set the Skysphere Effect to each part of the Skysphere model
foreach (ModelMesh mesh in SkySphere.Meshes)
{
    foreach (ModelMeshPart part in mesh.MeshParts)
    {
        part.Effect = SkySphereEffect;
    }
}

```

6. In your [Game.Draw](#), draw your scene as normal.

C#

```

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    //Draw the model, a model can have multiple meshes, so loop
    foreach (ModelMesh mesh in Model.Meshes)
    {
        //This is where the mesh orientation is set,
        //as well as our camera and projection
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.EnableDefaultLighting();
            effect.World = Matrix.Identity *
                Matrix.CreateRotationY(ModelRotation) *
                Matrix.CreateTranslation(ModelPosition);
            //effect.View = myCamera.View;
            effect.View = myCamera.ViewMatrix;
            effect.Projection = projectionMatrix;
        }
        //Draw the mesh, will use the effects set above.
        mesh.Draw();
    }
}

```

7. Draw the SkySphere by setting parameters to the SkySphere [Effect](#).
8. Draw the SkySphere [Model](#).

In this case we set the View and Projection matrices.

C#

```
// Set the View and Projection matrix for the effect
SkySphereEffect.Parameters["ViewMatrix"].SetValue(
    myCamera.ViewMatrix);
SkySphereEffect.Parameters["ProjectionMatrix"].SetValue(
    projectionMatrix);
// Draw the sphere model that the effect projects onto
foreach (ModelMesh mesh in SkySphere.Meshes)
{
    mesh.Draw();
}
```

9. Because this shader sets some render states before it runs, reset the affected render states to the values we want to use for further rendering.

C#

```
// Undo the renderstate settings from the shader
GraphicsDevice.RenderState.CullMode =
    CullMode.CullCounterClockwiseFace;
GraphicsDevice.RenderState.DepthBufferWriteEnable = true;
base.Draw(gameTime);
```

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect](#)

[TextureCube](#)

[Effect](#)

How To: Draw a Shadow

Demonstrates how to draw a shadow using [Matrix.CreateShadow](#) and the stencil buffer.

To draw a shadow, you use the shadow matrix to draw the original object without lighting. This sample demonstrates how to set up the shadow matrix and how to use the stencil buffer to avoid an uneven shadow.

Note

To render the scene, this sample uses the [BasicEffect](#) class. For a discussion of [BasicEffect](#) see [How To: Use BasicEffect](#). This sample also uses the **Quad** class introduced in [How To: Draw a Textured Quad](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StencilShadow_Sample.zip](#).

Drawing a Shadow

To draw a shadow

1. Use [PreferredDepthStencilFormat](#) to choose a depth buffer format that has some bits reserved for stencil buffering.

You can do this by checking each depth stencil format with [CheckDepthStencilMatch](#), and picking the best depth/stencil format for your needs.

C#

```
GraphicsDeviceManager graphics;
SpriteBatch spriteBatch;

public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
    graphics.PreferredDepthStencilFormat = SelectStencilMode();
}
private static DepthFormat SelectStencilMode()
{
    // Check stencil formats
    GraphicsAdapter adapter = GraphicsAdapter.DefaultAdapter;
    SurfaceFormat format = adapter.CurrentDisplayMode.Format;
    if (adapter.CheckDepthStencilMatch(DeviceType.Hardware, format,
        format, DepthFormat.Depth24Stencil8))
        return DepthFormat.Depth24Stencil8;
    else if (adapter.CheckDepthStencilMatch(DeviceType.Hardware, format,
        format, DepthFormat.Depth24Stencil8Single))
        return DepthFormat.Depth24Stencil8Single;
    else if (adapter.CheckDepthStencilMatch(DeviceType.Hardware, format,
        format, DepthFormat.Depth24Stencil4))
        return DepthFormat.Depth24Stencil4;
    else if (adapter.CheckDepthStencilMatch(DeviceType.Hardware, format,
        format, DepthFormat.Depth15Stencil1))
        return DepthFormat.Depth15Stencil1;
    else
        throw new InvalidOperationException(
            "Could Not Find Stencil Buffer for Default Adapter");
}
```

2. Create a [Plane](#) that represents the surface the shadow falls upon.

In this case, the shadow falls upon a textured quad drawn with the **Quad** class. Thus, we create a plane using three vertices of the quad.

C#

```
Quad wall;
Plane wallPlane;
protected override void Initialize()
{
    ...
    // Create a new Textured Quad to represent a wall
    wall = new Quad(Vector3.Zero, Vector3.Backward, Vector3.Up, 7, 7);
    // Create a Plane using three points on the Quad
    wallPlane =
        new Plane(wall.UpperLeft, wall.UpperRight, wall.LowerLeft);

    base.Initialize();
}
```

3. Use [CreateShadow](#) to create a shadow matrix based on the direction of your key light and the plane on which the shadow falls.

In this case, it is the [Plane](#) you created previously from a textured quad. The shadow matrix also contains a translation to make sure it does not intersect the quad. If the shadow used exactly the same plane as an object, the two objects would both be displayed intermittently. This is called the z-fighting effect.

C#

```
Vector3 shadowLightDir;
Matrix shadow;
protected override void LoadContent()
{
    ...
    shadowLightDir = quadEffect.Directionallight0.Direction;
    // Use the wall plane to create a shadow matrix,
    // and make the shadow slightly higher than the wall.
    // The shadow is based on the strongest light
    shadow = Matrix.CreateShadow(shadowLightDir, wallPlane) *
        Matrix.CreateTranslation(wall.Normal / 100);
    ...
}
```

4. In your game's [Draw](#) method, first draw your scene as normal.

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    // Draw floor, and draw model
    DrawQuad();

    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.EnableDefaultLighting();

            effect.View = View;
            effect.Projection = Projection;
        }
    }
}
```



```

        effect.World = modelWorld;
    }
    mesh.Draw();
}

```

5. Draw the shadow using a stencil buffer.

The stencil buffer ensures that the shadow appears one-dimensional. Our first step is to set [StencilEnable](#) to **true** and [Clear](#) the stencil buffer.

C#

```

// Draw shadow, using the stencil buffer to
// prevent drawing overlapping polygons

// Clear stencil buffer to zero.
GraphicsDevice.Clear(ClearOptions.Stencil, Color.Black, 0, 0);
GraphicsDevice.RenderState.StencilEnable = true;

```

6. Set up the stencil buffer test for the shadow by setting the [StencilFunction](#) to [CompareFunction.Equal](#) with a [ReferenceStencil](#) of 0.

This means you only draw a pixel when the stencil buffer value is 0. Setting [StencilPass](#) to [StencilOperation.Increment](#) means that after a pixel is drawn, the stencil buffer value becomes 1, so no more pixels will be drawn.

C#

```

// Draw on screen if 0 is the stencil buffer value
GraphicsDevice.RenderState.ReferenceStencil = 0;
GraphicsDevice.RenderState.StencilFunction =
    CompareFunction.Equal;
// Increment the stencil buffer if we draw
GraphicsDevice.RenderState.StencilPass =
    StencilOperation.Increment;

```

7. Set up the alpha blending for the shadow by setting [AlphaBlendEnable](#) to **true**, and setting the [SourceBlend](#) and [DestinationBlend](#) modes to use the alpha value.

C#

```

// Setup alpha blending to make the shadow semi-transparent
GraphicsDevice.RenderState.AlphaBlendEnable = true;
GraphicsDevice.RenderState.SourceBlend =
    Blend.SourceAlpha;
GraphicsDevice.RenderState.DestinationBlend =
    Blend.InverseSourceAlpha;

```

8. Draw the model again, this time as a shadow.

To make a shadow, we turn the ambient light color to black, disable other lights, and set an alpha transparency to make the ground below the shadow show through. The shape of the shadow is created by multiplying the model's world matrix with the shadow matrix.

C#

```

// Draw the shadow without lighting
foreach (ModelMesh mesh in model.Meshes)
{
    foreach (BasicEffect effect in mesh.Effects)
    {
        effect.AmbientLightColor = Vector3.Zero;
        effect.Alpha = 0.5f;
    }
}

```

```
        effect.DirectionalLight0.Enabled = false;
        effect.DirectionalLight1.Enabled = false;
        effect.DirectionalLight2.Enabled = false;
        effect.View = View;
        effect.Projection = Projection;
        effect.World = modelWorld * shadow;
    }
    mesh.Draw();
}
```

9. Reset the render states to normal, disabling stencil testing and alpha blending.

C#

```
// Return render states to normal

// turn stencilling off
GraphicsDevice.RenderState.StencilEnable = false;
// turn alpha blending off
GraphicsDevice.RenderState.AlphaBlendEnable = false;
```

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Use BasicEffect](#)

[How To: Draw a Textured Quad](#)

Reference

[CreateShadow](#)

[AlphaBlendEnable](#)

[StencilEnable](#)

[PreferredDepthStencilFormat](#)

How To: Use EffectParameters and EffectTechniques

Demonstrates how to use [EffectParameters](#) objects to interact with effect variables, and to use [EffectTechniques](#) to access different techniques in an [Effect](#).

In this example, a wrapper class is created to expose [EffectParameters](#) and [EffectTechniques](#) for the custom effect. The example shows you how to use the [EffectParameters](#) and [EffectTechniques](#). This class is used in the [How To: Implement Shadow Mapping](#) sample.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ShadowMapping_Sample.zip.](#)

Using EffectParameters and EffectTechniques

To Use EffectParameters

1. In your game's [LoadContent](#) method, load your [Effect](#) from the Content Manager.
2. Create an [EffectParameter](#) instance for each parameter in your effect that will be accessed during [Draw](#) or [Update](#).

You can use the [Parameters](#) indexed property on [Effect](#) to access any effect parameter, but this is slower than using [EffectParameters](#). For this reason, you should create an [EffectParameter](#) for each effect parameter that changes frequently.

C#

```
public EffectParameter mWorld;
public EffectParameter mCameraView;
public EffectParameter CameraPos;
public EffectParameter mCameraProj;
```

3. Use the [Parameters](#) indexed property on [Effect](#) to access the effect parameter, and then assign it to your [EffectParameter](#).

C#

```
mWorld = effect.Parameters["g_mWorld"];
mCameraView = effect.Parameters["g_mCameraView"];
CameraPos = effect.Parameters["g_CameraPos"];
mCameraProj = effect.Parameters["g_mCameraProj"];
```

4. Call [SetValue](#) on [EffectParameter](#) whenever you want to set a new value for the effect parameter.

You must call [SetValue](#) before calling [Begin](#) on your [Effect](#). If you change the value of an [EffectParameter](#) after calling [Begin](#) and before calling [End](#), you must call [CommitChanges](#) for the new value to take effect.

C#

```
MyEffect.CameraPos.SetValue(CameraPos);
MyEffect.mCameraView.SetValue(view);
MyEffect.mCameraProj.SetValue(projection);
MyEffect.LightPos.SetValue(LightPos);
MyEffect.mLightView.SetValue(Matrix.CreateLookAt(LightPos,
    bounds.Center, Vector3.Up));
```

To Use EffectTechniques

1. In your game's [LoadContent](#) method, load your [Effect](#) from the Content Manager.
2. Create an [EffectTechnique](#) instance for each technique in your [Effect](#) that you will use during [Update](#) or [Draw](#).

You can use the [Techniques](#) indexed property on [Effect](#) to access any effect technique, but this is slower than using

[EffectTechniques](#). For this reason, you should create an [EffectTechnique](#) for each effect technique that you use.

C#

```
public EffectTechnique texture;  
public EffectTechnique shadows;  
public EffectTechnique shadowMap;
```

3. Use the [Techniques](#) indexed property on [Effect](#) to access the effect technique, and then assign it to your [EffectTechnique](#).

C#

```
texture = effect.Techniques["TextureRender"];  
shadowMap = effect.Techniques["ShadowMapRender"];  
shadows = effect.Techniques["ShadowRender"];
```

4. Set the [CurrentTechnique](#) property on your [Effect](#) to the technique you wish to use when drawing.

You can use your [Effect](#) multiple times per [Draw](#) with different techniques.

C#

```
private void DrawScene(EffectTechnique technique)  
{  
    MyEffect.mWorld.SetValue(terrainWorld);  
    MyEffect.MeshTexture.SetValue(terrainTex);  
    foreach (ModelMesh mesh in terrain.Meshes)  
    {  
        foreach (Effect effect in mesh.Effects)  
        {  
            effect.CurrentTechnique = technique;  
            mesh.Draw();  
        }  
    }  
}
```

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Render a Model](#)

Reference

[Effect](#)

[CurrentTechnique](#)

[Techniques](#)

[Parameters](#)

[EffectTechnique](#)

[EffectParameter](#)

How To: Create a Depth Texture

Demonstrates how to create a texture that contains depth information for a scene using a customized `RenderTarget2D`, `DepthStencilBuffer`, and a simple Effect.

To render the depth to a texture, you create a new `RenderTarget2D` and a `DepthStencilBuffer` with your desired depth format. Then you render your scene by using a shader that draws each pixel in the render target as a depth value instead of a normal color. Finally, you use `GetTexture` on the `RenderTarget2D` to save that information to a `Texture2D`.

Note

To render the scene, this sample uses a technique from a customized Effect file. For more information, see [How To: Use Effect Parameters and Effect Techniques](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ShadowMapping_Sample.zip](#).

Creating a Depth Texture

To create a depth texture

1. In your game's `LoadContent` method, create a new `RenderTarget2D` for rendering the depth in your scene.

You may want to choose a surface format that gives you the most depth information. In this example, you will use `SurfaceFormat.Single` if the game computer supports it. Use `CheckDeviceFormat` on the computer to see if your chosen `SurfaceFormat` is supported on your game computer. Otherwise, consult [Xbox 360 Surface Formats](#) for supported `SurfaceFormats` on Xbox 360.

`SurfaceFormat.Single` creates a 32-bit floating point value for each pixel, representing the red channel. No bits are used for the green, blue, or alpha channels. Using floating point values for depth allows more precision for the shadow calculations. This results in smoother shadows. Using a large render target and enabling antialiasing for the render target also improves the quality of the shadow rendering.

To create the render target, there are four variables to consider: the proposed width of the texture, the proposed height of the texture, the surface format of the texture, and whether to use anti-aliasing. Not all PC video cards will accept all configurations of these four parameters. `CreateRenderTarget` tries to find the best match by first using `CheckDeviceFormat` to determine if the caller's preferred surface format is supported (in this case, a floating point texture). Next, it uses `CheckDeviceMultiSampleType` to confirm the anti-aliasing setting with the chosen surface format.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
```

```

{
    // Fall back to current display format
    surface = device.DisplayMode.Format;
}
// Or it can't accept that surface format
// with the current AA settings
else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
    DeviceType.Hardware, surface,
    device.PresentationParameters.IsFullScreen, type))
{
    // Fall back to no antialiasing
    type = MultiSampleType.None;
}

int width, height;

// See if we can use our buffer size as our texture
CheckTextureSize(device.PresentationParameters.BackBufferWidth,
    device.PresentationParameters.BackBufferHeight,
    out width, out height);

// Create our render target
return new RenderTarget2D(device,
    width, height, numberLevels, surface,
    type, 0);
}

```

Lastly, before creating the texture, **CreateRenderTarget** calls **CheckTextureSize**, which uses [GraphicsDeviceCapabilities](#) to see if the video card supports the chosen texture size (some video cards expect all textures to be powers of two, or square, or both). **CheckTextureSize** adjusts the texture size to fit the listed capabilities.

C#

```

public static bool CheckTextureSize(int width, int height,
    out int newwidth, out int newheight)
{
    bool retval = false;

    GraphicsDeviceCapabilities Caps;
    Caps = GraphicsAdapter.DefaultAdapter.GetCapabilities(
        DeviceType.Hardware);

    if (Caps.TextureCapabilities.RequiresPower2)
    {
        retval = true; // Return true to indicate the numbers changed

        // Find the nearest base two log of the current width,
        // and go up to the next integer
        double exp = Math.Ceiling(Math.Log(width)/Math.Log(2));
        // and use that as the exponent of the new width
        width = (int)Math.Pow(2, exp);
        // Repeat the process for height
        exp = Math.Ceiling(Math.Log(height)/Math.Log(2));
        height = (int)Math.Pow(2, exp);
    }
    if (Caps.TextureCapabilities.RequiresSquareOnly)
    {
        retval = true; // Return true to indicate numbers changed
        width = Math.Max(width, height);
    }
}

```

```

        height = width;
    }

    newwidth = Math.Min(Caps.MaxTextureWidth, width);
    newheight = Math.Min(Caps.MaxTextureHeight, height);
    return retval;
}

```

- Next in your `LoadContent` method, create a new `DepthStencilBuffer` for your custom `RenderTarget2D`.

The `DepthStencilBuffer` settings for width, height, and multisample quality should be the same as the values chosen for your render target. You may want to choose a depth format that gives you the most depth information. In this example, you will use `DepthFormat.Depth24Stencil8Single` if the game computer supports it. Use `CheckDepthStencilMatch` on Windows-based computers to see if your chosen `DepthFormat` is supported. Otherwise, consult [Xbox 360 Surface Formats](#) for supported `DepthFormats` on Xbox 360. `DepthFormat.Depth24Stencil8Single` creates a 24-bit floating-point value for depth. This allows more depth precision than the normal 24-bit fixed-point buffer. Still, you must be careful with floating-point values to guard against floating-point errors for depth values that are nearly identical.

C#

```

shadowDepthBuffer =
    GfxComponent.CreateDepthStencil(shadowRenderTarget,
    DepthFormat.Depth24Stencil8Single);

```

C#

```

public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target)
{
    return new DepthStencilBuffer(target.GraphicsDevice, target.Width,
        target.Height, target.GraphicsDevice.DepthStencilBuffer.Format,
        target.MultiSampleType, target.MultiSampleQuality);
}
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target, DepthFormat depth)
{
    if (GraphicsAdapter.DefaultAdapter.CheckDepthStencilMatch(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        target.Format,
        depth))
    {
        return new DepthStencilBuffer(target.GraphicsDevice,
            target.Width, target.Height, depth,
            target.MultiSampleType, target.MultiSampleQuality);
    }
    else
        return CreateDepthStencil(target);
}

```

- In your game's `Update` method, when you calculate a projection matrix from the point of view of the light source, include as many objects in the scene as possible if they are visible to the camera.

In this example, both objects are always visible to the camera and do not move, so the bounding sphere used is the bounding sphere of both objects combined.

- Set the near and far plane of the projection matrix as close as possible to the objects in the scene.

This will result in more accurate depth values for the depth map. The projection matrix is passed to the effect.

C#

```
protected override void Update(GameTime gameTime)
{
    ...
    Matrix proj =
        CalcLightProjection(LightPos, bounds, defaultViewport);
    MyEffect.mLightProj.SetValue(proj);
    base.Update(gameTime);
}
```

- In your `Draw` method, use `CompareFunction.LessEqual` to set the appropriate `DepthBufferFunction` for your depth texture effect.

When you depth-test a pixel, `CompareFunction.LessEqual` preserves lower depth values while discarding any pixels with a depth value higher than the current lowest value.

C#

```
GraphicsDevice.RenderState.DepthBufferFunction =
    CompareFunction.LessEqual;
```

- In your `Draw` method, call `GraphicsDevice.SetRenderTarget` to set the current render target (target 0) to the render target you created in Step 1.
- Make a copy of the current `DepthStencilBuffer` on the `GraphicsDevice` before assigning the `DepthStencilBuffer` you created previously to the `DepthStencilBuffer` property on `GraphicsDevice`.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
// Cache the current depth buffer
DepthStencilBuffer old = GraphicsDevice.DepthStencilBuffer;
// Set our custom depth buffer
GraphicsDevice.DepthStencilBuffer = shadowDepthBuffer;
```

- Render your scene using an effect that will draw the depth value of each pixel to your render target.

The following code shows a simple vertex and pixel shader for such an effect.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

- After you have rendered the depth values to the render target, call `SetRenderTarget` again

This sets the current render target (target 0) to **null**. Also, it resets the current render target to the display buffer.

- Set the `DepthStencilBuffer` on the `GraphicsDevice` to its former value.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
// Reset the depth buffer
GraphicsDevice.DepthStencilBuffer = old;
```

- Call `GetTexture` on your `RenderTarget2D` to get a `Texture2D` containing the depth values for your scene.

This is your depth texture.

C#


```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

Rendering Depth in HLSL

To render depth in HLSL

1. Use a custom vertex shader to render depth values in HLSL.

The vertex shader returns two values to the pixel shader. The first value is a POSITION that transforms the incoming POSITION into the view and projection space of the light source. The second value is the depth value of the transformed POSITION. The depth is calculated by dividing the z coordinate by the w coordinate. Dividing by w gives you a depth between 0 and 1. The depth is subtracted from 1 to get more precision from the floating point format. The depth is packed into a TEXCOORD semantic (in this case, TEXCOORD0) to be returned by the pixel shader.

```
struct VS_SHADOW_OUTPUT
{
    float4 Position : POSITION;
    float Depth : TEXCOORD0;
};
float4 GetPositionFromLight(float4 position)
{
    float4x4 WorldViewProjection =
        mul(mul(g_mWorld, g_mLightView), g_mLightProj);
    return mul(position, WorldViewProjection);
}
VS_SHADOW_OUTPUT RenderShadowMapVS(float4 vPos: POSITION)
{
    VS_SHADOW_OUTPUT Out;
    Out.Position = GetPositionFromLight(vPos);
    // Depth is Z/W. This is returned by the pixel shader.
    // Subtracting from 1 gives us more precision in floating point.
    Out.Depth.x = 1-(Out.Position.z/Out.Position.w);
    return Out;
}
```

2. Use a custom pixel shader to render depth values in HLSL.

The pixel shader returns one value, the depth of the pixel. This is calculated by the pixel shader and passed in the TEXCOORD0 semantic.

The depth is returned as the red value by the pixel shader. If you are using [SurfaceFormat.Single](#), this holds a 32-bit floating point value. More bits give you smoother shadows. Floating point will create smoother shadows than fixed point. However, this shader will create shadows using almost any SurfaceFormat value.

```
float4 RenderShadowMapPS( VS_SHADOW_OUTPUT In ) : COLOR
{
    // The depth is Z divided by W. We return
    // this value entirely in a 32-bit red channel
    // using SurfaceFormat.Single. This preserves the
    // floating-point data for finer detail.
    return float4(In.Depth.x,0,0,1);
}
```

See Also

Concepts

[3D Graphics Overview](#)

[What Is a Depth Buffer?](#)

[What Is a Depth Texture?](#)

[What Is a Render Target?](#)

Xbox 360 Surface Formats

Tasks

[How To: Draw a Model with a Custom Effect](#)

[How To: Use EffectParameters and EffectTechniques](#)

[How To: Implement Shadow Mapping](#)

Reference

[RenderTarget2D](#)

[GetTexture](#)

[SurfaceFormat](#)

[CheckDeviceFormat](#)

[Texture2D](#)

[DepthStencilBuffer](#)

[DepthFormat](#)

[CheckDepthStencilMatch](#)

[CheckDeviceMultiSampleType](#)

[GraphicsDeviceCapabilities](#)

[DepthBufferFunction](#)

[CompareFunction](#)

[GraphicsDevice.DepthStencilBuffer](#)

How To: Implement Shadow Mapping

Demonstrates how to use [depth textures](#) (that is, shadow maps) to create dynamic shadows in a scene. These shadows move when a light source moves, and are accurate against both flat and irregular surfaces.

To create shadows with shadow mapping, first you need to create a depth texture showing the depth of all objects in the scene from the point of view of the light source. Once you render pixels in your final scene from the camera's point of view, compare the distance from the pixel to the light source with the depth value encoded in the shadow map. When the depth of the rendered pixel is higher than the value in the shadow map, that pixel is in shadow.

Note

To render the scene, this sample uses a technique from a customized Effect file. For more information, see [How To: Draw a Model with a Custom Effect](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ShadowMapping_Sample.zip](#).

Working With Shadow Mapping

To create a shadow map effect

1. Load a custom effect that supports shadow mapping.

In this example, the effect has two major techniques: one for creating a shadow map, and the other for rendering a scene using the shadow map to draw shadows. A simple third technique is used to draw the light source. The effect is encapsulated in its own class.

C#

```
MyEffect = ShadowMapEffect.LoadEffect(Content);
```

2. In your game's [LoadContent](#) method, load the [Models](#) and [Effect](#) for your scene.
3. Remap the Models to use your custom [Effect](#) according to [How To: Draw a Model with a Custom Effect](#).
4. In [LoadContent](#), use the techniques described in [How To: Create a Depth Texture](#) to create a [RenderTarget2D](#) and [DepthStencilBuffer](#) for your shadow map.
5. In your game's [Update](#) method, set the position of your light source and your camera as parameters to the [Effect](#).

You must also set the view and projection matrices for both the shadow map and the camera itself.

C#

```
MyEffect.CameraPos.SetValue(CameraPos);
MyEffect.mCameraView.SetValue(view);
MyEffect.mCameraProj.SetValue(projection);
MyEffect.LightPos.SetValue(LightPos);
MyEffect.mLightView.SetValue(Matrix.CreateLookAt(LightPos,
    bounds.Center, Vector3.Up));
```

6. In your game's [Draw](#) method, use the techniques described in [How To: Create a Depth Texture](#) to create a shadow map for the scene.

C#

```
map = CreateShadowMap();
```

7. After creating the depth texture in [Draw](#), set it to an appropriate sampler value in your shadow mapping effect.

C#

```
// Set our shadow map into our effect
MyEffect.ShadowMapTexture.SetValue(map);
```

8. Draw your scene using a shader that supports shadow mapping.

The shader will apply ambient light only where the light casts a shadow.

C#

```
// Render our scene
GraphicsDevice.Clear(Color.CornflowerBlue);
DrawScene(MyEffect.shadows);
```

To render shadows in HLSL with a shadow map

- Use per-pixel lighting to render shadows with a shadow map in HLSL.

The vertex shader for this effect performs a normal world-view-projection transformation on the POSITION supplied by the video card. It also transforms the NORMAL. Both of these values are passed as output. The UV coordinate for the texture is passed through to the pixel shader, along with a copy of the untransformed POSITION value.

```
VS_OUTPUT RenderShadowsVS(
    float3 position : POSITION,
    float3 normal : NORMAL,
    float2 vTexCoord0 : TEXCOORD0 )
{
    VS_OUTPUT Output;

    //generate the world-view-projection matrix
    float4x4 wvp = mul(mul(g_mWorld, g_mCameraView), g_mCameraProj);

    //transform the input position to the output
    Output.Position = mul(float4(position, 1.0), wvp);

    //transform the normal to world space
    Output.vNormal = mul(normal, g_mWorld);

    //do not transform the position needed for the
    //shadow map determination
    Output.vPos = float4(position,1.0);

    //pass the texture coordinate as-is
    Output.TextureUV = vTexCoord0;

    //return the output structure
    return Output;
}
```

The first task for the pixel shader is calculating the normal lighting equation for the pixel. In this example, only diffuse lighting is calculated.

```
PS_OUTPUT RenderShadowsPS( PS_INPUT In )
{
    PS_OUTPUT Output;

    // Standard lighting equation
    float4 vTotalLightDiffuse = float4(0,0,0,1);
    float3 lightDir = normalize(g_LightPos-In.vPos); // direction of light
```

```
vTotalLightDiffuse += g_LightDiffuse * max(0,dot(In.vNormal, lightDir));
vTotalLightDiffuse.a = 1.0f;
```

Next, the pixel shader gets the position of this pixel on the shadow map. This is necessary because the shadow map is rendered from the light's perspective, and this pixel is rendered from the camera's perspective. The result of `GetPositionFromLight` is a screen coordinate. The screen coordinate is then converted into a UV coordinate so it can be accessed from the sampler containing the shadow map.

```
// Now, consult the ShadowMap to see if we're in shadow
float4 lightingPosition
    = GetPositionFromLight(In.vPos);// Get our position on the shadow map

// Get the shadow map depth value for this pixel
float2 ShadowTexC =
    0.5 * lightingPosition.xy / lightingPosition.w + float2( 0.5, 0.5 );
ShadowTexC.y = 1.0f - ShadowTexC.y;
```

Next, the shadow depth of this pixel is accessed using the `tex2D` function to access the shadow map. Then the depth of this pixel is determined by dividing the Z value of the position with the W value. This is subtracted from 1.0 before comparison because the shadow depths are similarly subtracted from 1.0. For more information, see `CreateDepthTexture`.

```
float shadowdepth = tex2D(ShadowMapSampler, ShadowTexC).r;

// Check our value against the depth value
float ourdepth = 1 - (lightingPosition.z / lightingPosition.w);
```

Now the shadow depth and true depth are compared. Floating-point precision errors can cause a banding effect if the two values are compared directly. The shadow depth is decreased slightly to ensure that pixels that should be lighted are lighted. If this pixel is indeed in shadow, the diffuse portion of the lighting is reset to black.

```
// Check the shadowdepth against the depth of this pixel
// a fudge factor is added to account for floating-point error
if (shadowdepth-0.03 > ourdepth)
{
    // we're in shadow, cut the light
    vTotalLightDiffuse = float4(0,0,0,1);
};
```

Finally, the total lighting equation is computed by adding the diffuse and ambient components together. This is the output returned from the pixel shader.

```
Output.RGBColor = tex2D(MeshTextureSampler, In.TextureUV) *
    (vTotalLightDiffuse + g_LightAmbient);

return Output;
}
```

See Also

Concepts

[3D Graphics Overview](#)

[What Is a Depth Buffer?](#)

[What Is a Depth Texture?](#)

[What Is a Render Target?](#)

Tasks

[How To: Draw a Model with a Custom Effect](#)

[How To: Use EffectParameters and EffectTechniques](#)

[How To: Create a Depth Texture](#)

Reference

[Model](#)

[Effect](#)

[RenderTarget2D](#)

[DepthStencilBuffer](#)

"What Is" Articles

Provides brief overviews of some 3D graphics concepts.

In This Section

[What Is Antialiasing?](#)

Antialiasing is the process of softening hard edges in an image so that polygons appear less jagged.

[What Is Color Blending?](#)

Color blending is the process of mixing two colors together to produce a third color.

[What Is a Depth Buffer?](#)

A depth buffer is a buffer that is the same width and height as your render target. This buffer records the depth of each pixel that is rendered.

[What Is a Depth Texture?](#)

A depth texture is a texture that contains the data from the [depth buffer](#) for a particular scene.

[What Is a Model Bone?](#)

A model bone is a matrix that represents the position of a mesh relative to other bones in a 3D model.

[What Is a Render Target?](#)

A render target is a buffer where the video card draws pixels for a scene that is being rendered by an [Effect Class](#).

[What Is a Stencil Buffer?](#)

A stencil buffer is similar to a [depth buffer](#). In fact, it uses part of the depth buffer (for this reason, the depth buffer is often called the *depth-stencil buffer*). The stencil buffer allows the programmer to set a stencil function that will test the "reference" stencil value - a global value - against the value already in the stencil buffer each time a pixel is rendered.

[What Is Texture Mapping?](#)

Texture mapping is the process of determining where in a particular space a texture will be applied.

What Is Antialiasing?

Antialiasing is the process of softening hard edges in an image so that polygons appear less jagged.

Antialiasing is normally accomplished by *multisampling*, which means you use more than one pixel ("sample") from an image to determine the color of a pixel in your final image. The more samples per pixel, the smoother the resulting image.

Both PC and Xbox 360 platforms support Full Screen Antialiasing (FSAA) modes, where extra pixels are rendered to a render target and used as samples to create an antialiased final image. 4x FSAA uses four samples per pixel by rendering the scene to a render target with twice the height and width of the backbuffer. It uses the extra samples in the render target to create an antialiased final image. 2x FSAA is also common on PCs and supported by the Xbox 360.

Use the [CheckDeviceMultiSampleType](#) method to query for antialiasing support on your game machine. Also, use the [MultiSampleQuality](#) and [MultiSampleType](#) properties of the [PresentationParameters](#) to select an antialiasing mode for your backbuffer. For more information, see [How To: Enable Antialiasing \(Multisampling\)](#). You can set [PreferMultiSampling](#) on the [GraphicsDeviceManager](#) to **true** to let the [GraphicsDeviceManager](#) choose the antialiasing mode.

You can use the [RenderState](#) property [MultiSampleAntiAlias](#) to enable or disable multisampling at run time.

See Also

Concepts

[3D Graphics Overview](#)

[What Is a Render Target?](#)

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[CheckDeviceMultiSampleType](#)

[PreferMultiSampling](#)

[MultiSampleQuality](#)

[MultiSampleType](#)

What Is Color Blending?

Color blending is the process of mixing two colors together to produce a third color.

The first color is called the *source color*. This is the new color being added. The second color is called the *destination color*. This is the color that already exists (in the backbuffer, for example). Each color has a separate blend factor that determines how much of each color is combined into the final product. Once the source and destination colors have been multiplied by their blend factors, the results are combined according to the specified blend function. The normal blend function is simple addition.

The full formula looks like this:

$$(source \times source \text{ blend factor}) \text{ (blend function) } (destination \times destination \text{ blend factor})$$

The source blend factor is specified by the [SourceBlend](#) property, and the destination blend factor is specified by the [DestinationBlend](#) property. The [BlendFunction](#) property specifies the blend function to use, normally [BlendFunction.Add](#). In that case the formula looks like this:

$$(source \times \mathbf{SourceBlend}) + (destination \times \mathbf{DestinationBlend})$$

When the [AlphaBlendEnable](#) property is **false**, no blending occurs during rendering. In this case, the source pixel overwrites the destination pixel. When [AlphaBlendEnable](#) is **true**, you can create a lot of special effects using the [SourceBlend](#) and [DestinationBlend](#) properties:

Blend Type	Blend Settings
Alpha Blending	$(source \times \text{Blend.SourceAlpha}) + (destination \times \text{Blend.InvSourceAlpha})$
Additive Blending	$(source \times \text{Blend.One}) + (destination \times \text{Blend.One})$
Multiplicative Blending	$(source \times \text{Blend.Zero}) + (destination \times \text{Blend.SourceColor})$
2X Multiplicative Blending	$(source \times \text{Blend.DestinationColor}) + (destination \times \text{Blend.SourceColor})$



Figure 1. This picture illustrates four common blend modes. From left to right: Alpha blending, Additive blending, Multiplicative blending, and 2X Multiplicative blending. The top image in each column is the source image and below, it's effect when added to the destination.

Alpha blending uses the alpha channel of the source color to create a transparency effect so that the destination color appears through the source color. For example, if you clear your backbuffer to [Color.Gray](#), it will be colored (0.5,0.5,0.5,1). If you then take a [Color.White](#) color with a partial alpha value (1,1,1,0.4), the result will be 60 percent of the destination color and 40 percent of the source: $(0.5 \times 0.6) + (1 \times 0.4)$. The resulting color will be (0.7,0.7,0.7, 1). The alpha values are multiplied as well - $(.6 \times 1) + .4$ gives us an alpha value of 1.

In some cases, alpha blending is automatic. When drawing sprites using the [SpriteBatch](#) class, choosing [SpriteBlendMode.AlphaBlend](#) configures alpha blending for you.

By default, the alpha channel is blended along with the red, green, and blue channels using the [SourceBlend](#) and [DestinationBlend](#) properties. You can choose to customize the blending for just the alpha channel by using the [AlphaSourceBlend](#) and [AlphaDestinationBlend](#) properties. When you compute the alpha channel, these properties are used as blending factors if the [SeparateAlphaBlendEnabled](#) property is **true**.

See Also

Concepts

[2D Graphics Overview](#)

[3D Graphics Overview](#)

Tasks

[How To: Draw a Masked Sprite over a Background](#)

Reference

[RenderState](#)

[SourceBlend](#)

DestinationBlend
AlphaSourceBlend
AlphaDestinationBlend
Blend
SpriteBlendMode.AlphaBlend

What Is a Depth Buffer?

A depth buffer is a buffer that is the same width and height as your render target. This buffer records the depth of each pixel that is rendered.

When a pixel is rendered a second time - such as when one object is rendered behind another - the depth buffer will keep either the previous depth value, or replace it with the depth value for the second pixel. Which depth is preserved and which depth is discarded depends on the depth function you select. For example, if [CompareFunction.LessEqual](#) is the current depth function, depth values that are less than or equal to the current value are preserved. Any value greater than the current depth value is discarded. This is called the *depth test*. The depth test occurs every time a pixel is rendered. When a pixel passes the depth test, its color is written to the render target and its depth is written to the depth buffer.

The depth of a pixel is determined based on the size of the view and projection matrix selected for rendering. A pixel that touches the near plane of the projection has depth 0. A pixel that touches the far plane of the projection has depth 1. As each object in the scene is rendered, normally the pixels that are closest to the camera are kept, as those objects block the view of the objects behind them.

The depth buffer may also contain stencil bits - for this reason it's often called the *depth-stencil buffer*. The depth format describes the composition of the depth buffer. The depth buffer is always 32 bits, but those bits can be arranged in different ways, similar to how texture formats can vary. A common depth format is `Depth32`, where all 32 bits are reserved for depth information. Another common format is `DepthFormat.Depth24Stencil8`, where 24 bits are used for depth calculation and 8 bits are used by the stencil buffer. `DepthFormat.Depth24Stencil8Single` is a more unusual format where the 24 bits for the depth buffer are arranged as a floating point value. Use the `AutoDepthStencilFormat` property on `PresentationParameters` to set the default depth format.

Use `RenderState.DepthBufferEnable` to enable or disable depth buffering. Use `RenderState.DepthBufferFunction` to change the comparison function used for the depth test. The depth buffer can be cleared separately by passing `ClearOptions.DepthBuffer` to the `GraphicsDevice.Clear` method. Use the `DepthStencilBuffer` class to create your own depth buffer. [How To: Create a Depth Texture](#) provides an example of how to create a custom `DepthStencilBuffer`.

See Also

Concepts

[3D Graphics Overview](#)

[What Is a Depth Texture?](#)

[What Is a Stencil Buffer?](#)

Tasks

[How To: Create a Depth Texture](#)

Reference

[DepthStencilBuffer](#)

[GraphicsDevice.Clear](#)

[RenderState.DepthBufferEnable](#)

[RenderState.DepthBufferFunction](#)

[DepthFormat](#)

What Is a Depth Texture?

A depth texture is a texture that contains the data from the [depth buffer](#) for a particular scene.

The color of each pixel in the texture represents the depth of the polygon under that pixel, taken from the depth buffer. You can use depth textures to render accurate shadows using a technique called shadow mapping. For this reason, depth textures are also known as *shadow maps*.

There are two common ways to encode the depth information into a texture. The first is to use a standard RGBA surface format for your texture, and treat all four colors as one 32-bit buffer and copy the depth into that buffer (depth formats normally assign 24 or 32 bits for depth data). The second is to use a floating-point surface format for your texture combined with a floating-point depth buffer. Depth buffers are normally fixed-point. When using the floating-point approach, it is more accurate to subtract the depth from 1.0 before storing it, due to the way floating-point numbers are encoded.

Both of these techniques require a custom vertex and pixel shader. [How To: Create a Depth Texture](#) provides an example of a depth texture shader and the code necessary to use it.

See Also

Concepts

[3D Graphics Overview](#)

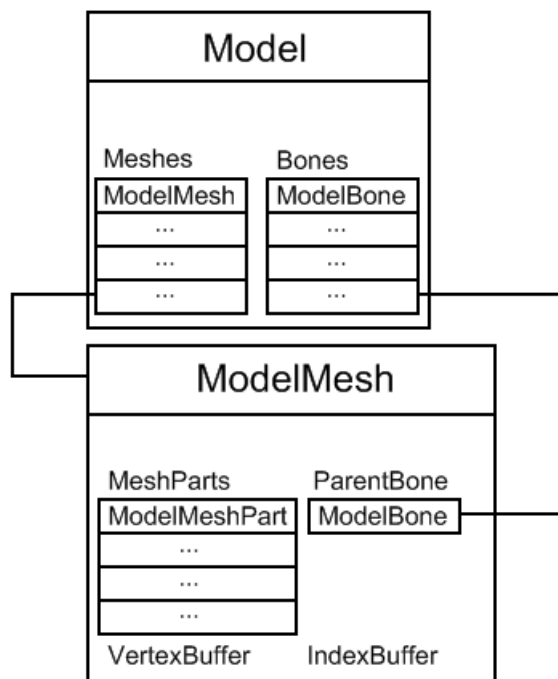
Tasks

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

What Is a Model Bone?

A model bone is a matrix that represents the position of a mesh relative to other bones in a 3D model.



In the XNA Framework, the [Model](#) class represents the whole model. The [Model](#) contains a [ModelMesh](#) for each separate mesh in the model. Each [ModelMesh](#) contains a [ParentBone](#), which controls the mesh's position and orientation relative to the model. The [Model](#) has a [Root](#) bone, which determines the model's position and orientation. Every [ModelBone](#) can have one parent and many children. The [Root](#) bone on the [Model](#) object is the ultimate parent. Its children are bones on [ModelMesh](#) objects—objects which might have other [ModelMesh](#) bones as their children, and so on. In any given family of bones, rotating the parent bone also rotates the children, and their children, and so on.

Every bone has a transformation matrix (called [Transform](#)) that defines its position and rotation relative to the position of the parent bone. This rotation and translation applies to all the vertices in the [ModelMesh](#) (for example, all the vertices that connect to that bone). To animate a bone, you multiply the default bone transform by a new matrix. When you draw the [ModelMesh](#), you then base your world matrix on the bone's transform.

The easiest way to incorporate transformed bones into drawing is to use the [CopyAbsoluteBoneTransformsTo](#) method. This method takes the bone transforms, which are relative to each other, and iterates over them to make them relative to the [Root](#) bone of the [Model](#). Then it returns a copy of these transforms. When you draw each [ModelMesh](#), you can use the absolute bone transform as the first part of your world matrix. This way you won't have to worry about parent bones and their relationships.

See Also

Concepts

[3D Graphics Overview](#)

Tasks

[How To: Render a Model](#)

Reference

[ModelBone](#)

[Model](#)

[Model.CopyAbsoluteBoneTransformsTo Method](#)

[ModelMesh](#)

What Is a Render Target?

A render target is a buffer where the video card draws pixels for a scene that is being rendered by an [Effect Class](#).

The default render target is called the back buffer - this is the part of video memory that contains the next frame to be drawn. You can create other render targets with the [RenderTarget2D](#) class - in effect, reserving new regions of video memory for drawing. Most games render a lot of content to other render targets besides the back buffer ("offscreen"), then assemble the different graphical elements in stages, combining them to create the final product in the back buffer.

A render target has a width and height. The width and height of the back buffer are the final resolution of your game (although on Xbox 360 that final result is scaled to match the user's screen). An offscreen render target does not need to have the same width and height as the back buffer. Small parts of the final image can be rendered in small render targets, then copied to another render target later. A render target also has a surface format, which describes how many bits are allocated to each pixel and how they are divided between red, green, blue, and alpha. For example, [SurfaceFormat.Bgr32](#) allocates 32 bits per pixel: 8 bits for each color and 8 bits for the alpha channel. As an option, render targets can perform antialiasing on all the images that are rendered into them.

To use a render target, create a [RenderTarget2D](#) object with the width, height, and other options you prefer. Then call [GraphicsDevice.SetRenderTarget](#) to make your render target the current render target. From this point on, any **Draw** calls you make will draw into your render target. When you are finished with the render target, call [GraphicsDevice.SetRenderTarget](#) to a new render target (or null for the back buffer). Then at any time you can call [RenderTarget2D.GetTexture](#) to get the contents of the render target for further processing.

Render targets work in conjunction with the depth-stencil buffer. If you set a new render target, it will use the existing depth-stencil buffer. If the new render target has different multisampling settings than the depth-stencil buffer, or a larger width and height, you will need a new depth-stencil buffer to match. You must also use a depth format in your depth-stencil buffer that is compatible with the surface format of your render target.

You can sometimes render to more than one render target at the same time. The number of simultaneous render targets your graphics device supports is given by the [MaxSimultaneousRenderTargets](#) property. There are numerous caveats for using multiple render targets. For more information, see [Render Targets](#).

See Also

Concepts

[3D Graphics Overview](#)

[Displays, Client Bounds, Viewports, and Back Buffers](#)

[What Is a Depth Buffer?](#)

[What Is a Stencil Buffer?](#)

Tasks

[How To: Apply a Pixel Shader to Sprites](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[RenderTarget](#)

[RenderTarget2D.GetTexture](#)

[GraphicsDevice.SetRenderTarget](#)

What Is a Stencil Buffer?

A stencil buffer is similar to a [depth buffer](#). In fact, it uses part of the depth buffer (for this reason, the depth buffer is often called the *depth-stencil buffer*). The stencil buffer allows the programmer to set a stencil function that will test the "reference" stencil value - a global value - against the value already in the stencil buffer each time a pixel is rendered.

The outcome of that stencil test determines if the color value of the pixel is written to the render target, and if the depth value of that pixel is written to the depth buffer.

For example, if you render some objects to a scene while the reference stencil is 0, and the stencil buffer has been cleared to 1, the stencil buffer will contain a cut-out pattern of zeros where those objects were rendered. If the reference value was then set to 1, and the [StencilFunction](#) was set to [CompareFunction.LessEqual](#), the only pixels that could be rendered to the scene are those that are not in the same location where the stencil value was set to 0. This is a basic way to use the stencil buffer to create an area that is off-limits to the current rendering pass.

The stencil buffer can be used in more sophisticated ways. It is possible to specify [StencilOperations](#) that go beyond replace or discard, to increment or decrement the stencil buffer after each successful stencil test. This can be combined with the [StencilMask](#) value to ensure that the stencil test only operates on a part of the stencil buffer.

To use the stencil buffer, the [DepthFormat](#) must reserve some bits for the stencil buffer. The [DepthFormat.Depth24Stencil8](#) depth format allows 8 bits for a stencil buffer. When you combine this with the [RenderState.StencilMask Property](#), this can provide eight different stencil buffers for the programmer. The [DepthFormat.Depth24Stencil4](#) depth format uses 4 bits for the stencil buffer, while the [DepthFormat.Depth15Stencil1](#) format only allows one bit. The stencil buffer can be cleared separately by passing [ClearOptions.StencilBuffer](#) to the [GraphicsDevice.Clear](#) method.

You can use the [DepthStencilBuffer](#) class to create your own depth-stencil buffer. You may need to create your own depth-stencil buffer when you create a custom render target

See Also

Concepts

[3D Graphics Overview](#)

[What Is a Depth Buffer?](#)

[What Is a Render Target?](#)

Tasks

[How To: Draw a Shadow](#)

[How To: Create a Depth Texture](#)

Reference

[DepthStencilBuffer](#)

[StencilFunction](#)

[StencilMask](#)

[StencilOperation](#)

[DepthFormat](#)

What Is Texture Mapping?

Texture mapping is the process of determining where in a particular space a texture will be applied.

A texture consists of a series of pixels (also called *texels*), each occupying a texture coordinate determined by the width and height of the texture. These texture coordinates are then mapped into values ranging from 0 to 1 along a u and v axes (u is width, v is height). This process is called UV mapping. The resulting coordinates are UV coordinates.

Whenever you render a polygon, a UV coordinate is calculated for each vertex in the triangle. These three UV coordinates dictate how a texture is applied to the polygon by the pixel shader. When the u or v values move below 0 or above 1, the polygon is applied according to the texture address mode that has been set. When [TextureAddressMode.Border](#) is in effect, any pixels where the UV mapping is outside the 0-1 range are colored with the border color. When [TextureAddressMode.Clamp](#) is in effect, the color of the nearest pixel is used. [TextureAddressMode.Wrap](#) indicates that the texture should repeat across the triangle. When you specify [TextureAddressMode.Mirror](#), the texture wraps, but reverses itself when it crosses a UV boundary. Texture-addressing modes are specified for each axis separately, so you could wrap along the u-axis while clamping the v-axis.

If the texture is too large or too small for the polygon, the texture must be filtered to fit the space. There are two types of filtering that can be applied to textures: magnification and minification. A magnification filter enlarges a texture to fit a polygon. A minification filter reduces the texture to fit into a smaller area. Texture magnification is normally straightforward, and results in a blurrier image. Texture minification is more complicated, and improper minification can result in aliasing – that is, jagged edges.

The most popular approach to minification is to create mipmaps for each texture. A mipmap is a pre-shrunk texture, normally half the size of the original. The mipmap itself then gets mipmapped, and this process continues until a 1x1 texture is created. This is the final mipmap for the texture. You can think of mipmaps as a chain, starting with the original texture and becoming smaller and smaller until the 1 texel texture is reached. When minification is needed, first the appropriate mipmapped texture is chosen, then that mipmap is applied to the object, with real-time texture filtering if needed. The default Texture processor for the Content Pipeline has an option to generate mipmaps automatically.

XNA supports five texture filters. [TextureFilter.Point](#) uses the nearest corresponding point on the texture, with no filtering. [TextureFilter.Linear](#) uses bilinear interpolation to sample four neighboring texels and create an average value. [TextureFilter.PyramidalQuad](#) also uses four samples and favors texels nearer to the center of the resulting pixel when calculating the final result. The [TextureFilter.GaussianQuad](#) works the same way, using a different mathematical weighting - similar to a bell curve where the top of the curve is the center of the resulting pixel. [TextureFilter.Anisotropic](#) is a filtering method designed for surfaces that are not facing the camera (such as the ground near a horizon).

See Also

Concepts

[3D Graphics Overview](#)

[2D Graphics Overview](#)

Tasks

[How To: Tile a Sprite](#)

Reference

[TextureAddressMode](#)

[TextureFilter](#)

Math

Provides classes and methods for manipulating vectors and matrices. This section discusses math and includes examples of common math-related game tasks.

In This Section

[Math Overview](#)

Overview of the math-related functionality provided by the XNA Framework.

[Collision Detection Overview](#)

Overview of the collision-detection-related functionality provided by the XNA Framework.

[How To: Transform a Point with a Matrix](#)

This example demonstrates how to use the [Vector3](#) and [Matrix](#) classes to transform a point.

[How To: Rotate and Move a Camera](#)

Demonstrates how to rotate and move a first-person camera in a 3D environment.

[How To: Make a First-Person Camera](#)

Demonstrates how to create a first-person camera.

[How To: Make a Third-Person Camera](#)

Demonstrates how to create a third-person camera.

[How To: Script the Camera to Follow a Curve](#)

Demonstrates how to use the [Curve](#) and [CurveKey](#) classes to script the movement of the camera.

[How To: Position the Camera to View All Objects in a Scene](#)

Demonstrates how to position the camera so that all objects in a scene are within the view frustum while maintaining the camera's original orientation.

[How To: Detect Whether Two Models Collide](#)

Demonstrates how to use the [BoundingSphere](#) class to check whether two models are colliding.

[How To: Detect Whether a User Clicked a 3D Object](#)

Demonstrates how to check whether the mouse is positioned over a 3D object by creating a ray starting at the camera's near clipping plane and ending at its far clipping plane.

See Also

Concepts

[Math Content Catalog at XNA Creators Club Online](#)

Math Overview

The XNA Framework Math Libraries can be found in the [Microsoft.Xna.Framework](#) namespace, alongside a number of additional types that deal with the XNA Framework Application model.

- [Coordinate system](#)
- [Mathematical Constants and Scalar Manipulation](#)
- [Basic Geometric Types](#)
- [Precision and Performance](#)

Coordinate system

The XNA Framework uses a right-handed coordinate system, with the positive z-axis pointing toward the observer when the positive x-axis is pointing to the right, and the positive y-axis is pointing up.

Mathematical Constants and Scalar Manipulation

The XNA Framework provides the [MathHelper Members](#) class for [manipulating scalar values](#) and retrieving some [common mathematical constants](#). This includes methods such as the [ToDegrees](#) and [ToRadians](#) utility methods for converting between degrees and radians.

Basic Geometric Types

The XNA Framework Math library has multiple basic geometric types that can be used to manipulate objects in 2D or 3D space. The primitive objects in this library represent the data required to represent a geometric object or an operation on that object. Each geometric type has a number of mathematical operations that are supported for the type.

Vectors

The XNA Framework provides the [Vector2](#), [Vector3](#) and [Vector4](#) classes for representing and manipulating vectors. A vector is typically used to represent a direction and magnitude. However, in the XNA framework it might also be used to store a coordinate or some other data type with the same storage requirements.

Each vector class has methods for performing standard vector operations such as:

- [Dot product](#)
- [Cross product](#)
- [Normalization](#)
- [Transformation](#)
- [Linear](#), [Cubic](#), [Catmull-Rom](#), or [Hermite spline](#) interpolation.

Matrices

The XNA Framework provides the [Matrix](#) class for transformation of geometry. The [Matrix](#) class uses a row major order to address matrices, which means that the row is specified before the column when describing an element of a two-dimensional matrix. The [Matrix](#) class provides methods for performing standard matrix operations such as calculating the [determinate](#) or [inverse](#) of a matrix, in addition to helper methods for creating scale, translation, and rotation matrices.

Quaternions

The XNA Framework provides the [Quaternion](#) structure to represent and calculate the efficient rotation about a vector around a specified angle.

Curves

The [Curve](#) class represents a hermite curve that allows you to interpolate varying positions at different times without having to explicitly define each position. The curve is defined by a collection of [CurveKey](#) points representing each varying position at different times.

This class can be used not only for spatial motion, but also to represent any response that changes over time.

Bounding Volumes

The XNA Framework provides the [BoundingBox](#), [BoundingFrustum](#), [BoundingSphere](#), [Plane](#), and [Ray](#) classes for representing simplified versions of geometry for the purpose of efficient collision and hit-testing. These classes have methods for checking for intersection and containment with each other.

Precision and Performance

The XNA Framework Math libraries are single-precision. This means that the primitives and operations contained in this library use 32-bits to represent floating-point numbers in order to achieve a balance between precision and efficiency when performing large numbers of calculations.

The 32-bit floating-point numbers can represent a range of values from negative $3.402823e38$ to positive $3.402823e38$. The 32-bits of a floating-point number are used to store the sign, mantissa, and exponent of the number. The result is seven digits of floating-point precision. Some numbers, such as π , $1/3$, or the square root of two, can only be approximated with a limited amount of memory. Because the number of bits used to represent a decimal number limit the range of numbers it can represent, floating-point numbers only approximate decimal numbers that require more than seven digits of precision. Consequently, rounding errors can occur under different conditions through the use of any binary-representation approximating a floating point number.

You can find more information about single-precision numbers in the documentation for the [Single](#) data type.

★Best Practice

Depending on the hardware platform your game is targeting, you may have multiple processors available to perform floating-point calculations. For example, on the Xbox 360 platform and on many Windows-based computers, you have a CPU, on which your game code is running, but you also have GPU, on which your shader code is running. Therefore, you can also perform floating point calculations on the GPU using shaders and [HLSL intrinsic functions](#). GPUs are very fast at performing large numbers of floating-point operations. This can be very useful, for example, if you need to perform a large number of floating point operations to animate a particle system.

However, you need to remember that the Zune platform does not have a GPU and does not support shaders. For Zune, you can use a conditional compilation directive `#if ZUNE` to write downlevel-code that behaves gracefully on this platform while still taking advantage of the performance gains on other types of hardware.

See Also

Concepts

[Math Content Catalog at XNA Creators Club Online](#)

Collision Detection Overview

Overview of the collision-detection–related functionality provided by the XNA Framework.

Collision detection is the process of determining whether objects in a game world overlap each other. The XNA Framework provides several classes and methods to speed implementation of collision detection systems in games.

- [Bounding Volume Classes](#)
- [Non-Bounding Volume Classes](#)
- [Contains and Intersects Methods](#)
- [Adding New Collision Data Structures](#)

Bounding Volume Classes

The XNA Framework has three classes that represent three-dimensional areas in the world. You can use the bounding volume classes to approximate the volume occupied by models with shapes that are less expensive to do collision checks with. All of the bounding volume classes support intersection and containment tests with each other and the plane and ray classes.

Bounding Sphere

The [BoundingSphere Structure](#) represents the space occupied by a sphere.

There are several benefits of using a bounding sphere for collision detection.

- Sphere to sphere checks are very fast. To check for collision between two spheres, the distance between the centers of the spheres is compared to the sum of the radii of both spheres. If the distance is less than the combined radii of both spheres, the spheres intersect.
- The [BoundingSphere Structure](#) class is compact. It stores only a vector representing its center and its radius.
- Unlike a bounding box, a bounding sphere doesn't need to be recreated if the model rotates. If the model being bounded rotates, the bounding sphere will still be large enough to contain it.
- Moving a bounding sphere is inexpensive. Just add a value to the center.

There is one major drawback to using the bounding sphere class for collision detection.

- Unless the object being approximated is sphere shaped, the bounding sphere will have some empty space, which could result in false positive results. Long narrow objects will have the most empty space in their bounding spheres.

Bounding Box

The [BoundingBox Structure](#) represents the space occupied by a box. The bounding box class is axis aligned. Each face of the bounding box is perpendicular to the x-axis, the y-axis, or the z-axis.

There are several benefits of using the bounding box for collision detection.

- The bounding box class fits rectangular shapes aligned with the axis very well. Compared to the bounding sphere class, the bounding box class provides a much tighter fit for non-rotated rectangular objects.
- Because the bounding box class is axis aligned, you can make certain assumptions that result in collision checks between bounding boxes being quicker than a bounding box that can be rotated.

There are a few drawbacks of using the bounding box for collision detection.

- Rotating a bounding box causes it to no longer be axis aligned. Because of this, if you rotate a model being bounded, you will need to recreate the bounding box. Doing so can be slow, since all the points in an object are iterated through to get the bounding box. If the model has not changed orientation, you can translate the bounding box instead of recreating it.
- If the model being bounded is not aligned to the axis, the bounding box will have some empty space. The amount of empty space will be greatest when the object is rotated 45 degrees from an axis.
- Empty space in the bounding box can result in false positives when checking for collision.

Bounding Frustum

You can use [BoundingFrustum Class](#) to create a bounding volume that corresponds to the space visible to the camera. You create a bounding frustum from the combined view and projection matrix that the camera is using currently. If the camera moves or rotates, you need to recreate the bounding frustum. The bounding frustum isn't used to determine when two objects collide, but rather when an object is in collision with the volume of space viewable by the camera. Objects that do not intersect

and are not contained by the bounding frustum are not visible to the camera and don't need to be drawn. For complex models, this can save the graphics card a lot of work.

Non-Bounding Volume Classes

Plane

The [Plane Structure](#) describes a plane in space. The plane is defined by a vector perpendicular to the plane and a point on the plane. The plane class supports intersection tests with the bounding volume classes. The plane class's intersection test returns the tested object's position relative to the plane. The return value indicates whether the object intersects the plane. If the object does not intersect the plane, the return value indicates whether the object is on the plane's front side or back side.

Ray

The [Ray Structure](#) describes a ray starting at a point in space. The ray structure supports intersection tests with the bounding volume classes. The return value of the ray intersection tests is the distance the intersection occurred at, or null if no intersection occurred.

Model

In addition to the information needed to draw a model, the [Model Class](#) contains bounding volumes for its parts. When a model is imported, the content pipeline calculates the bounding sphere for each of the model's parts. To check for collision between two models, you can compare the bounding spheres for one model to all of the bounding spheres of the other model.

Contains and Intersects Methods

Bounding volume classes have methods to support two types of collision tests: intersection tests and containment tests. The intersects methods check whether the two objects being tested overlap in any way. As soon as the test finds that the objects do intersect, it returns without trying to determine the degree of the intersection. The contains methods determine whether the objects simply intersect or whether one of the objects is completely contained by the other. Since the intersects methods need only determine whether an intersection occurred, they tend to be faster than the contains methods. Use the contains methods only when the degree of intersection is relevant.

Adding New Collision Data Structures

When implementing other bounding volume classes and intersection tests, you will probably need to add a custom content pipeline processor. For example, your game might need to use convex hulls for collision detection. You could use a custom processor to determine the convex hull and then place it in the model's tag field. Then, when the model is loaded at run time, the convex hull information would be available in the model. For more information, see [Extending an XNA Framework Standard Processor](#).

See Also

Tasks

[FuelCell: "Ships" Passing in the Night](#)

[XNA Creators Club Online](#)

[Collision Content Catalog at XNA Creators Club Online](#)

How To: Transform a Point with a Matrix

This example demonstrates how to use the [Vector3](#) and [Matrix](#) classes to transform a point.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download TransformPoint_Sample.zip](#).

Transforming a Point with a Matrix

To transform a point

1. Create a [Matrix](#) by using [CreateRotationY](#) or one of the other **Create** methods.
2. Pass the point and the [Matrix](#) to the [Vector3.Transform](#) method.

C#

```
static Vector3 RotatePointOnYAxis(Vector3 point, float angle)
{
    // Create a rotation matrix that represents a rotation of angle radians.
    Matrix rotationMatrix = Matrix.CreateRotationY(angle);

    // Apply the rotation matrix to the point.
    Vector3 rotatedPoint = Vector3.Transform(point, rotationMatrix);

    return rotatedPoint;
}
```

See Also

Matrix Creation Methods

[CreateRotationX](#)

[CreateRotationY](#)

[CreateRotationZ](#)

[CreateScale](#)

[CreateTranslation](#)

How To: Rotate and Move a Camera

Demonstrates how to rotate and move a first-person camera in a 3D environment. You can rotate the camera about its y-axis, and move it forward and backward. You control the camera's position and orientation by using the directional keys on your keyboard or by using the D-pad of your Xbox 360 gamepad.

Note

This sample is the base for a more advanced sample that implements three camera types: first-person, first-person (offset), and third-person. For more information about this sample, see [How To: Make a Third-Person Camera](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download FirstPersonCamera_Sample.zip](#).

Overview

This sample is based on several assumptions.

- The camera will move frequently, so the camera view [Matrix](#) is created and set every time [Game.Update](#) is called.
- The projection [Matrix](#) may also change frequently for effects such as zooming.
- You have added a model to the project as described in [How To: Render a Model](#).

For the sake of simplicity, the sample limits the camera object to an axis of rotation (y-axis) and forward and backward movement. This simplifies the process (and resultant code) for rendering the sample 3D scene. The following steps show you how to render the sample scene.

To render the sample scene

1. Determine the location and orientation of the camera object.
2. To create a view matrix as part of the standard world-view-projection matrix trilogy of 3D rendering using the camera position, orientation (also referred to as the "look at point"), and the world space's up vector, call [CreateLookAt](#).
3. Create a perspective matrix that determines the near and far clipping planes and the aspect of the projection with a call to [CreatePerspectiveFieldOfView](#).
4. In the [Draw](#) method of your game, initialize a [BasicEffect](#) object with the transformational matrices created earlier (world, view projection), and then render all existing 3D models.

Rotating and Moving a Camera

To rotate and move the camera

1. Determine the camera's position in world coordinates.
2. Determine the reference [Vector3](#) to which the rotation of the camera is relative.

The direction should not change during the game, and usually it will be (0, 0, 1) or (0, 0, -1).

C#

```
// Set the direction the camera points without rotation.
Vector3 cameraReference = new Vector3(0, 0, 1);
```

3. Create a rotation [Matrix](#) determining the current amount of rotation for the camera object.

Because the camera is limited to one axis of rotation, this matrix represents the rotation of the camera around its own y-axis. Use [CreateRotationY](#) to create a rotation [Matrix](#) representing the rotation around the y-axis.

C#

```
Matrix rotationMatrix = Matrix.CreateRotationY(avatarYaw);
```

- Use [Transform](#) and the rotation [Matrix](#) to transform a copy of the reference [vector](#).

This represents the direction the camera is pointing in transformed (or view) space.

C#

```
// Create a vector pointing the direction the camera is facing.
Vector3 transformedReference = Vector3.Transform(cameraReference, rotationMatrix);
```

- Add the camera's current position to the transformed direction [vector](#).

The result is the position to which the camera is pointing.

C#

```
// Calculate the position the camera is looking at.
Vector3 cameraLookat = cameraPosition + transformedReference;
```

- Create a new view [Matrix](#) using [CreateLookAt](#).

- Use [CreateLookAt](#) to pass the camera's current position and the transformed direction vector.

The third parameter of [CreateLookAt](#) is the up direction of the camera. Typically, it is [Vector3.Up](#) (0, 1, 0). This matrix [Matrix](#) controls how world coordinates are transformed to camera coordinates.

C#

```
// Set up the view matrix and projection matrix.
view = Matrix.CreateLookAt(cameraPosition, cameraLookat, new Vector3(0.0f, 1.0f, 0.0f)
);
```

- Use [CreatePerspectiveFieldOfView](#) to create a new projection [Matrix](#).

This [Matrix](#) controls how camera coordinate values are transformed to screen coordinates.

The first parameter is the field of view of the projection [Matrix](#) expressed in radians. A typical field of view of 45 degrees would be expressed as $\pi/4$ radians. The second parameter is the aspect ratio of the projection [Matrix](#); it corrects for the difference in width and height of a viewspace. The third and fourth parameters specify the near and far distances at which the objects will be visible.

C#

```
// Set distance from the camera of the near and far clipping planes.
static float nearClip = 1.0f;
static float farClip = 2000.0f;
Viewport viewport = graphics.GraphicsDevice.Viewport;
float aspectRatio = (float)viewport.Width / (float)viewport.Height;

proj = Matrix.CreatePerspectiveFieldOfView(viewAngle, aspectRatio, nearClip, farClip);
```

- Loop through each 3D model to be rendered as described in [How To: Render a Model](#) using the projection matrix and view matrix created above.

An identity matrix simplifies the code for the world matrix.

C#

```
void DrawModel(Model model, Matrix world, Texture2D texture)
{
    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (BasicEffect be in mesh.Effects)
        {
            be.Projection = proj;
            be.View = view;
        }
    }
}
```



```
        be.World = world;
        be.Texture = texture;
        be.TextureEnabled = true;
    }
    mesh.Draw();
}
```

See Also [How To: Render a Model](#)

How To: Make a First-Person Camera

Demonstrates how to create a first-person camera.

The example controls the camera by using the method shown in [How To: Rotate and Move a Camera](#), with an additional offset for the camera to place it where the game avatar's head should be. The example assumes a model has been added to the project as described in [How To: Render a Model](#).

Tip

This technique is implemented in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: Setting the Scene](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download FirstPersonCamera_Sample.zip](#).

Making a First-Person Camera

To make a first-person camera

1. Precalculate the camera's offset from the avatar.

The offset is used to place the camera roughly at the location of the avatar's head.

C#

```
Vector3 avatarHeadOffset = new Vector3(0, 10, 0);
```

2. Track the position and rotation of the avatar during gameplay.
3. Create a rotation [Matrix](#) using [CreateRotationY](#) and the avatar's current direction.

C#

```
Matrix rotationMatrix = Matrix.CreateRotationY(avatarYaw);
```

4. Use the rotation [Matrix](#) and [Vector3.Transform](#) to transform a copy of the camera's head offset.

C#

```
// Transform the head offset so the camera is
// positioned properly relative to the avatar.
Vector3 headOffset =
    Vector3.Transform(avatarHeadOffset, rotationMatrix);
```

5. Calculate the current position of the camera.

The position of the camera will be the position of the avatar, plus the avatar's transformed head offset.

C#

```
// Calculate the camera's current position.
Vector3 cameraPosition = avatarPosition + headOffset;
```

6. Use the rotation [Matrix](#) and [Vector3.Transform](#) to transform a copy of the camera's reference [vector](#).

C#

```
// Create a vector pointing the direction the camera is facing.
Vector3 transformedReference =
    Vector3.Transform(cameraReference, rotationMatrix);
```

- Calculate the position at which the camera is looking.

This "look-at" position will be the camera's position, plus the camera's transformed reference [vector](#).

C#

```
// Calculate the position the camera is looking at.
Vector3 cameraLookat = transformedReference + cameraPosition;
```

- Create a new view [Matrix](#).

You can create the new [Matrix](#) by passing the camera position and camera look-at to [CreateLookAt](#). The view [Matrix](#) controls how world coordinate values are transformed to camera coordinates.

C#

```
view = Matrix.CreateLookAt(cameraPosition, cameraLookat,
    new Vector3(0.0f, 1.0f, 0.0f));
```

- Use [CreatePerspectiveFieldOfView](#) to create a new projection [Matrix](#).

The projection [Matrix](#) controls how camera coordinate values are transformed to screen coordinates.

C#

```
Viewport viewport = graphics.GraphicsDevice.Viewport;
float aspectRatio = (float)viewport.Width / (float)viewport.Height;

proj = Matrix.CreatePerspectiveFieldOfView(viewAngle, aspectRatio,
    nearClip, farClip);
```

- Loop through each model in the world drawing it as described in [How To: Render a Model](#) using the projection matrix and view matrix created above.

For the world matrix, use [Matrix.CreateTranslation](#) and the object's current position in the world.

C#

```
void DrawBoxes()
{
    for (int z = 0; z < 9; z++)
    {
        for (int x = 0; x < 9; x++)
        {
            DrawModel(box,
                Matrix.CreateTranslation(x * 60, 0, z * 60),
                boxTexture);
        }
    }
}

/// <summary>
/// Draws the 3D specified model.
/// </summary>
/// <param name="model">The 3D model being drawn.</param>
/// <param name="world">Transformation matrix for world coords.
/// </param>
/// <param name="texture">Texture used for the drawn 3D model.
/// </param>
void DrawModel(Model model, Matrix world, Texture2D texture)
{
    foreach (ModelMesh mesh in model.Meshes)
    {
```

```
    foreach (BasicEffect be in mesh.Effects)
    {
        be.Projection = proj;
        be.View = view;
        be.World = world;
        be.Texture = texture;
        be.TextureEnabled = true;
    }
    mesh.Draw();
}
```

See Also [How To: Rotate and Move a Camera](#)

[How To: Render a Model](#)

How To: Make a Third-Person Camera

Demonstrates how to create a third-person camera.

The example controls the camera using the method shown in [How To: Rotate and Move a Camera](#), with an additional offset for the camera so it is looking down at the game avatar. Turning the avatar causes the camera to rotate around the avatar's position. The current camera type is displayed in the upper-left corner of the screen and updated whenever the TAB or left shoulder button is pressed.

The example assumes a model has been added to the project as described in [How To: Render a Model](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ThirdPersonCamera_Sample.zip](#).

Making a Third-Person Camera

To make a third-person camera

1. Decide on a camera offset [vector](#) from the avatar.

An offset of the form (0, N, -N) will put the camera above and behind the avatar at a 45-degree angle.

C#

```
Vector3 thirdPersonReference = new Vector3(0, 200, -200);
```

2. Track the position and rotation of the avatar.

Next, you need to create a rotation [Matrix](#) for the direction that the avatar is facing.

3. Create the rotation [Matrix](#) with [CreateRotationY](#), using the avatar's current rotation as its parameter.

C#

```
Matrix rotationMatrix = Matrix.CreateRotationY/avatarYaw);
```

4. Transform a copy of the camera offset [vector](#) by using the [Transform](#) method and the rotation [Matrix](#).

C#

```
// Create a vector pointing the direction the camera is facing.
Vector3 transformedReference =
    Vector3.Transform(thirdPersonReference, rotationMatrix);
```

5. Calculate the camera's position as the avatar's position, and calculate the transformed camera offset.

C#

```
// Calculate the position the camera is looking from.
Vector3 cameraPosition = transformedReference + avatarPosition;
```

6. Create a view [Matrix](#) with the [CreateLookAt](#) method.

The avatar's current position will be the *cameraTarget* parameter.

C#

```
view = Matrix.CreateLookAt(cameraPosition, avatarPosition,
    new Vector3(0.0f, 1.0f, 0.0f));
```

7. Create a new projection [Matrix](#) with [CreatePerspectiveFieldOfView](#).

The projection [Matrix](#) controls how camera coordinate values are transformed to screen coordinates.

C#

```
Viewport viewport = graphics.GraphicsDevice.Viewport;
float aspectRatio = (float)viewport.Width / (float)viewport.Height;

proj = Matrix.CreatePerspectiveFieldOfView(viewAngle, aspectRatio,
    nearClip, farClip);
```

8. Loop through each model in the world drawing it as described in [How To: Render a Model](#) using the projection matrix and view matrix created above.

For the world matrix, use [Matrix.CreateTranslation](#) and the object's current position in the world.

C#

```
void DrawBoxes()
{
    for (int z = 0; z < 9; z++)
    {
        for (int x = 0; x < 9; x++)
        {
            DrawModel(box,
                Matrix.CreateTranslation(x * 60, 0, z * 60),
                boxTexture);
        }
    }
}

/// <summary>
/// Draws the 3D specified model.
/// </summary>
/// <param name="model">The 3D model being drawn.</param>
/// <param name="world">Transformation matrix for world coords.
/// </param>
/// <param name="texture">Texture used for the drawn 3D model.
/// </param>
void DrawModel(Model model, Matrix world, Texture2D texture)
{
    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (BasicEffect be in mesh.Effects)
        {
            be.Projection = proj;
            be.View = view;
            be.World = world;
            be.Texture = texture;
            be.TextureEnabled = true;
        }
        mesh.Draw();
    }
}
```

See Also [How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Render a Model](#)

How To: Script the Camera to Follow a Curve

Demonstrates how to use the [Curve](#) and [CurveKey](#) classes to script the movement of the camera.

Using [Curves](#) allows a path to be defined by a small number of control points with the [Curves](#) calculating the points on the path between the control points.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ScriptedCamera_Sample.zip](#).

Scripting the Camera to Follow a Curve

To script camera movement

1. Create an instance of the [Curve](#) class for each component being scripted.

In this case, you need two sets of three curves. One is for each of the x, y, and z components of the camera's position, and the other is for the position at which the camera is looking (the "look-at" position).

C#

```
class Curve3D
{
    public Curve curveX = new Curve();
    public Curve curveY = new Curve();
    public Curve curveZ = new Curve();
    ...
}
Curve3D cameraCurvePosition = new Curve3D();
Curve3D cameraCurveLookat = new Curve3D();
```

2. Set the [PreLoop](#) and [PostLoop](#) type of each [Curve](#).

The [PreLoop](#) and [PostLoop](#) types determine how the curve will interpret positions before the first key or after the last key. In this case, the values will be set to [CurveLoopType.Oscillate](#). Values past the ends of the curve will change direction and head toward the opposite side of the curve.

C#

```
curveX.PostLoop = CurveLoopType.Oscillate;
curveY.PostLoop = CurveLoopType.Oscillate;
curveZ.PostLoop = CurveLoopType.Oscillate;

curveX.PreLoop = CurveLoopType.Oscillate;
curveY.PreLoop = CurveLoopType.Oscillate;
curveZ.PreLoop = CurveLoopType.Oscillate;
```

3. Add [CurveKeys](#) to the [Curves](#).
4. Specify the time each [CurveKey](#) should be reached and the camera position when the [CurveKey](#) is reached.

In this case, each point in time will have three [CurveKeys](#) associated with it – one for each of the x, y, and z coordinates of the point on the [Curve](#).

C#

```
public void AddPoint(Vector3 point, float time)
{
    curveX.Keys.Add(new CurveKey(time, point.X));
    curveY.Keys.Add(new CurveKey(time, point.Y));
```

```

        curveZ.Keys.Add(new CurveKey(time, point.Z));
    }
    void InitCurve()
    {
        float time = 0;
        cameraCurvePosition.AddPoint(new Vector3(7.5f, 0, -45), time);
        cameraCurveLookat.AddPoint(new Vector3(9, 0, 9), time);
        time += 2000;
        cameraCurvePosition.AddPoint(new Vector3(3f, 0, -36), time);
        time += 2000;
        cameraCurvePosition.AddPoint(new Vector3(12f, 0, -30), time);
        time += 2000;
        cameraCurvePosition.AddPoint(new Vector3(3f, 0, -24), time);
        time += 2000;
        cameraCurvePosition.AddPoint(new Vector3(12f, 0, -18), time);
        time += 2000;
        ...
        cameraCurvePosition.SetTangents();
        cameraCurveLookat.SetTangents();
    }

```

5. Loop through each [Curve](#) setting the [TangentIn](#) and [TangentOut](#) of each [CurveKey](#).

The tangents of the [CurveKeys](#) control the shape of the [Curve](#). Setting the tangents of the [CurveKeys](#) to the slope between the previous and next [CurveKey](#) will give a curve that moves smoothly through each point on the curve.

C#

```

public void SetTangents()
{
    CurveKey prev;
    CurveKey current;
    CurveKey next;
    int prevIndex;
    int nextIndex;
    for (int i = 0; i < curveX.Keys.Count; i++)
    {
        prevIndex = i - 1;
        if (prevIndex < 0) prevIndex = i;

        nextIndex = i + 1;
        if (nextIndex == curveX.Keys.Count) nextIndex = i;

        prev = curveX.Keys[prevIndex];
        next = curveX.Keys[nextIndex];
        current = curveX.Keys[i];
        SetCurveKeyTangent(ref prev, ref current, ref next);
        curveX.Keys[i] = current;
        prev = curveY.Keys[prevIndex];
        next = curveY.Keys[nextIndex];
        current = curveY.Keys[i];
        SetCurveKeyTangent(ref prev, ref current, ref next);
        curveY.Keys[i] = current;

        prev = curveZ.Keys[prevIndex];
        next = curveZ.Keys[nextIndex];
        current = curveZ.Keys[i];
        SetCurveKeyTangent(ref prev, ref current, ref next);
        curveZ.Keys[i] = current;
    }
}

```



```

}
static void SetCurveKeyTangent(ref CurveKey prev, ref CurveKey cur,
    ref CurveKey next)
{
    float dt = next.Position - prev.Position;
    float dv = next.Value - prev.Value;
    if (Math.Abs(dv) < float.Epsilon)
    {
        cur.TangentIn = 0;
        cur.TangentOut = 0;
    }
    else
    {
        // The in and out tangents should be equal to the
        // slope between the adjacent keys.
        cur.TangentIn = dv * (cur.Position - prev.Position) / dt;
        cur.TangentOut = dv * (next.Position - cur.Position) / dt;
    }
}
}

```

6. Add code to evaluate the x, y, and z coordinates of the [Curves](#) at any given time by passing the elapsed time to the [Evaluate](#) method of each of the [Curves](#).

C#

```

public Vector3 GetPointOnCurve(float time)
{
    Vector3 point = new Vector3();
    point.X = curveX.Evaluate(time);
    point.Y = curveY.Evaluate(time);
    point.Z = curveZ.Evaluate(time);
    return point;
}

```

7. Create a variable to track the amount of time that has passed since the camera started moving.

C#

```

double time;

```

8. In [Game.Update](#), set the camera's position and look-at position based on the elapsed time since the camera started moving, and then set the camera's view and projection matrices as in [How To: Rotate and Move a Camera](#).

C#

```

// Calculate the camera's current position.
Vector3 cameraPosition =
    cameraCurvePosition.GetPointOnCurve((float)time);
Vector3 cameraLookat =
    cameraCurveLookat.GetPointOnCurve((float)time);

```

9. In [Game.Update](#), use [gameTime.ElapsedGameTime.TotalMilliseconds](#) to increment the time since the camera started moving.

C#

```

time += gameTime.ElapsedGameTime.TotalMilliseconds;

```

See Also [How To: Rotate and Move a Camera](#)
[How To: Make a First-Person Camera](#)
[How To: Make a Third-Person Camera](#)

How To: Position the Camera to View All Objects in a Scene

Demonstrates how to position the camera so that all objects in a scene are within the view frustum while maintaining the camera's original orientation.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample..

[Download FitCameraToScene_Sample.zip.](#)

Positioning the Camera to View All Objects in a Scene

To position the camera to view all objects in a scene

1. Create a [BoundingSphere](#) class that contains all of the objects in the scene. To create the sphere, loop through all of the objects in the scene, merging the [BoundingSphere](#) classes that contain them with [BoundingSphere.CreateMerged](#).
2. If you are not already tracking the [BoundingSphere](#) classes for collision detection, use [CreateFromBoundingBox](#) or [CreateFromPoints](#) to create them from [BoundingBox](#) classes or points.

In this example, the [BoundingSphere](#) classes are created from [BoundingBox](#) classes.

C#

```
BoundingSphere GetSceneSphere()
{
    BoundingSphere sceneSphere =
        new BoundingSphere(new Vector3(.5f, 1, .5f), 1.5f);
    for (int z = 0; z < 5; z++)
    {
        for (int x = 0; x < 5; x++)
        {
            BoundingSphere boundingSphere =
                sphere.Meshes[0].BoundingSphere;
            boundingSphere.Center = new Vector3(x * 3, 0, z * 3);

            sceneSphere = BoundingSphere.CreateMerged(
                sceneSphere, boundingSphere);
        }
    }

    return sceneSphere;
}
```

3. Set the position of the camera to the center of the [BoundingSphere](#) that contains the scene.

C#

```
cameraPosition = sceneSphere.Center;
```

4. Determine the distance from the center of the [BoundingSphere](#) that the camera needs to be to view the entire scene.

This distance is equal to the hypotenuse of the triangle formed by the center of the sphere, the desired camera position, and the point where the sphere touches the view frustum. One angle of the triangle is known to be the field of view of the camera divided by two. One leg of the triangle is known to be the radius of the sphere. Given these two measurements, you can calculate the hypotenuse as the radius of the sphere divided by the sine of half the field of view.

C#

```
float distanceToCenter =
    sceneSphere.Radius / (float)Math.Sin(FOV / 2);
```

5. Get the [Backward](#) vector of the view [Matrix](#) and flip its X component.

C#

```
Vector3 back = view.Backward;  
back.X = -back.X; //flip x's sign
```

6. To move the camera backward with respect to its orientation, multiply the desired distance by the adjusted back vector from the previous step.

The camera is now facing the center of the sphere containing the scene and is far enough back that the sphere fits in the camera's view frustum.

C#

```
cameraPosition += (back * distanceToCenter);
```

See Also [How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

[How To: Render a Model](#)

How To: Detect Whether Two Models Collide

Demonstrates how to use the [BoundingSphere](#) class to check whether two models are colliding.

Tip

This technique is implemented in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: "Ships" Passing in the Night](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download CollisionBetweenSpheres_Sample.zip](#).

Detecting Whether Two Models Collide

To check whether two objects are colliding

1. Track the position of a model as it moves about the game world.

C#

```

struct WorldObject
{
    public Vector3 position;
    public Vector3 velocity;
    public Model model;
    public Texture2D texture2D;
    public Vector3 lastPosition;
    public void MoveForward()
    {
        lastPosition = position;
        position += velocity;
    }
    public void Backup()
    {
        position -= velocity;
    }
    public void ReverseVelocity()
    {
        velocity.X = -velocity.X;
    }
}

```

2. Make a nested loop with the first model's meshes as the outer loop and the second model's meshes as the inner loop.
3. Inside the loop, follow these steps.
 - a. Get the bounding sphere for the current mesh of the first model and the current mesh of the second model.
 - b. Offset the bounding spheres by the current positions of the models.
 - c. Call the [BoundingSphere.Intersects](#) method to check the pairs of bounding spheres for collision.

If the method returns **true**, the objects are colliding.
 - d. If the models are colliding, break out of the loop.

C#

```

static void CheckForCollisions(ref WorldObject c1, ref WorldObject c2)
{
    for (int i = 0; i < c1.model.Meshes.Count; i++)
    {

```

```
// Check whether the bounding boxes of the two cubes intersect.
BoundingBox c1BoundingBox = c1.model.Meshes[i].BoundingBox;
c1BoundingBox.Center += c1.position;

for (int j = 0; j < c2.model.Meshes.Count; j++)
{
    BoundingBox c2BoundingBox = c2.model.Meshes[j].BoundingBox;
    c2BoundingBox.Center += c2.position;

    if (c1BoundingBox.Intersects(c2BoundingBox))
    {
        c2.ReverseVelocity();
        c1.Backup();
        c1.ReverseVelocity();
        return;
    }
}
}
```

Note

For an example of determining a particle's path after it hits a surface, see [Vector3.Reflect](#).

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Render a Model](#)

Concepts

[Collision Detection Overview](#)

[Collision Content Catalog at XNA Creators Club Online](#)

How To: Detect Whether a User Clicked a 3D Object

Demonstrates how to check whether the mouse is positioned over a 3D object by creating a ray starting at the camera's near clipping plane and ending at its far clipping plane.

Note

This example applies only to Windows development. The [Mouse](#) and [MouseState](#) objects are not supported on Xbox 360.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download Click3DObject_Sample.zip.](#)

Detecting Whether a User Clicked a 3D Object

To check whether the mouse is positioned over a 3D object

1. Get the current state of the mouse by using [GetState](#).

C#

```
MouseState mouseState = Mouse.GetState();
```

2. Get the current screen coordinates of the mouse from [X](#) and [Y](#).

C#

```
int mouseX = mouseState.X;
int mouseY = mouseState.Y;
```

3. Using [Viewport.Unproject](#), determine points in world space on the near and far clip plane. For the point on the near plane, pass a source vector with x and y set to the mouse position, and z set to 0.
4. For the point on the far plane, pass a source vector with x and y set to the mouse position, and z set to 1.
5. For both points, pass [Unproject](#) the current projection matrix, the view matrix, and a translation matrix for the point (0,0,0).

C#

```
Vector3 nearsource = new Vector3((float)mouseX, (float)mouseY, 0f);
Vector3 farsource = new Vector3((float)mouseX, (float)mouseY, 1f);

Matrix world = Matrix.CreateTranslation(0, 0, 0);

Vector3 nearPoint = GraphicsDevice.Viewport.Unproject(nearsource,
    proj, view, world);

Vector3 farPoint = GraphicsDevice.Viewport.Unproject(farsource,
    proj, view, world);
```

6. Create a [Ray](#) starting at the near point and pointing toward the far point.

C#

```
// Create a ray from the near clip plane to the far clip plane.
Vector3 direction = farPoint - nearPoint;
direction.Normalize();
Ray pickRay = new Ray(nearPoint, direction);
```

7. Use [Intersects](#) to loop through the objects in the world to check whether the [Ray](#) intersects each object.

8. If the [Ray](#) intersects an object, check whether it is the closest object intersected so far. If it is, store the object and the distance at which it was intersected, replacing any previously stored object.
9. When you completely loop through the objects, the last object stored will be the closest object underneath the area the user clicked.

See Also [How To: Rotate and Move a Camera](#)

[How To: Render a Model](#)

Input

Provides classes and methods for retrieving user input for keyboard, mouse, and Xbox 360 controller devices.

In This Section

[Input Overview](#)

This article is an introduction to handling user input in XNA Game Studio.

[Button Mapping Between Different Controllers and the Xbox 360 Gamepad](#)

Discusses the various controller subtypes and the various mappings for each controller to the standard Xbox 360 gamepad.

[How To: Detect Whether a Controller Button Is Pressed](#)

Demonstrates how to detect whether a user has pressed a digital button on a connected Xbox 360 Controller.

[How To: Detect Whether a Controller Button Has Been Pressed This Frame](#)

Demonstrates how to detect whether a user has just pressed a digital button on a connected Xbox 360 Controller.

[How To: Detect Whether a Controller Is Disconnected](#)

Demonstrates how to detect whether an Xbox 360 Controller connected to the system at game start has become disconnected.

[How To: Get the Current Mouse Position \(Windows\)](#)

Demonstrates how to get the current horizontal and vertical position of the mouse cursor relative to the upper-left corner a game window.

[How To: Detect Whether a Key Is Pressed](#)

Demonstrates how to detect whether a user has pressed or released a key on the keyboard.

See Also

Concepts

[Input Content Catalog at XNA Creators Club Online](#)

Input Overview

Input is a general term referring to the process of receiving actions from the user. In XNA Game Studio, the [Microsoft.Xna.Framework.Input](#) namespace provides support for all input devices.

Note

APIs related to input devices physically not available in your game are always available to your code. For example, you can access the [Mouse](#) API on Zune, but you cannot retrieve the relevant mouse position. Also, you should not expect an exception or build error when you access APIs related to unavailable input devices.

Xbox 360 Controller, Mouse, and Keyboard

The Xbox 360 Controller, the keyboard, and the mouse are common input devices. What follows is the differences between these input devices and guidelines for how to use them with your game.

Digital vs. Analog

Input devices have two types of controls: digital and analog. Digital controls report only two possible states—on and off—and are represented by Boolean values. Examples of digital controls are the **START** button on the Xbox 360 Controller and any button on a keyboard.

Analog controls report a range of values, rather than just off and on. Examples of analog controls are the Xbox 360 Controller stick and the movements of the mouse. Analog values can be represented in a variety of ways. In XNA Game Studio, an analog value on the Xbox 360 Controller is represented as a floating-point value between -1.0 and 1.0 for the sticks, and between 0.0 and 1.0 for the triggers. For the mouse, XNA Game Studio reports mouse cursor values in pixels.

Polling and States

For all input devices in XNA Game Studio, input is received from the user by way of **polling**: each frame, call a function to receive the current state of an input device, and compare to previous states if necessary. A state is a snapshot of an input device's interactions from the user. At any given moment, a control may have a variety of buttons pressed, analog sticks or triggers held in a direction. The current positions of these buttons and analog controls are reported in a state structure. The following sections describe the methods for retrieving and using state for each input device.

Input Device Types

Each input device has specific advantages and disadvantages. The following table shows a brief comparison of three devices.

Input Device	Digital Buttons	Analog Controls	Vibration Effects	Supported on Windows	Supported on Xbox 360	Number Allowed on System
Xbox 360 Controller	14	4	Yes	Yes	Yes	4
Keyboard	> 100	0	No	Yes	Yes	1
Mouse	5	3	No	Yes	No	1

Xbox 360 Controller

The Xbox 360 Controller provides a good combination of digital buttons and analog sticks, so that many types of games can be played with it.

The controller can be used on either Windows or Xbox 360 systems. Up to four controllers are supported.

How to Use

1. Each frame of your game, call the static [GetState](#) method from the [GamePad](#) class.
2. Pass in the index of the player you wish to query. The player associated with a controller is visible on the Ring of Light on the controller.
3. Query the [GamePadState](#) structure you get back to retrieve information on the state of the controller. For more information on retrieving state, including how to determine whether the user has pressed or released a button since the last frame, see [How To: Detect Whether a Controller Button Is Pressed](#).

You may store a previous controller state if you wish to compare states to determine input changes from one frame to the next.

Note

Controllers may become disconnected during normal gameplay. For information on how to properly detect and handle this situation, see [How To: Detect Whether a Controller Is Disconnected](#).

Keyboard

The keyboard, with a multitude of digital buttons, is best for complicated simulations with many functions, or when typing text is necessary—but a keyboard has no analog controls for precise movement.

Not all keyboards support the entire range of buttons listed in [Keys](#); for example, older keyboards may not support the **VolumeDown**, **VolumeUp**, and **VolumeMute** members.

If the user's keyboard is connected via USB, it can be used on a Windows system, or on an Xbox 360.

How to Use

1. Each frame of your game, call the static [GetState](#) method from the [Keyboard](#) class.
2. Query the [KeyboardState](#) structure you get back to retrieve information on the state of the keyboard.

You may store a previous keyboard state if you wish to compare states to determine input changes from one frame to the next. See [How To: Detect Whether a Key Is Pressed](#) for more information.

Mouse

The mouse, with its precise analog control, is best for selecting objects, such as in a strategy game, or for moving a viewport, such as in a first-person action game—but a mouse has few digital buttons. Modern mouse devices have three extra buttons in addition to the standard left and right buttons: two digital buttons, usually on the side of the mouse, and a digital button that is activated by clicking down on the scroll wheel. The scroll wheel itself functions as an extra analog control in two directions, but does not provide precise input.

Not all mouse devices support all buttons; for example, older mouse devices may not support the [MiddleButton](#), [XButton1](#) and [XButton2](#) properties.

You can set the [Game.IsMouseVisible Property](#) to **true** to make the mouse cursor visible; use the default (**false**) if you are drawing your own cursor or don't want a cursor.

Mouse devices are used on Windows systems only. They are not supported on Xbox 360.

How to Use

1. Each frame of your game, call the static [GetState](#) method from the [Mouse](#) class.
2. Query the [MouseState](#) structure that is returned to retrieve information on the state of the mouse. Mouse coordinates are relative to the upper-left corner of the game window.

You may store a previous mouse state if you wish to compare states to determine input changes from one frame to the next.

For more information on mouse state, including an example, see [How To: Get the Current Mouse Position \(Windows\)](#).

Input on Zune

Zune devices have two digital buttons, and a ZunePad that is an analog control. The two buttons are the **Start** and **Back** buttons. The ZunePad features directional input that is retrieved with the [GamePad.GetState](#) method. The **Back** button maps to [GamePadState.Buttons.Back](#), and the **Play/pause** button maps to [GamePadState.Buttons.B](#).

For more information, see [Zune Programming Considerations](#).

See Also

Concepts

[Input Content Catalog at XNA Creators Club Online](#)

Button Mapping Between Different Controllers and the Xbox 360 Gamepad

Discusses the various controller subtypes and the various mappings for each controller to the standard Xbox 360 gamepad.

To see how various Zune buttons map to the buttons on a standard Xbox 360 gamepad, see [Zune Button Mapping](#).

- [Required and Optional Features](#)
- [Alternate Guitar](#)
- [Arcade Stick](#)
- [Big Button Pad](#)
- [Dance Pad](#)
- [Drum Kit](#)
- [Flight Stick](#)
- [Gamepad](#)
- [Guitar](#)
- [Wheel](#)

Required and Optional Features

Support for the following features is required from all controller subtypes:


- The action buttons: **A**, **B**, **X**, and **Y**.
- The **Back**, **Start**, and **Xbox Guide** buttons.
- The directional pad (Up, Down, Left and Right), the Ring of Light, and voice support.

Support for the following features is optional:

- The serial port, which is part of the Expansion Port.
- The **Left/Right** stick buttons (that is, thumbsticks).
- The **Left/Right** triggers.
- The **Left/Right** shoulder buttons (that is, bumpers).

Alternate Guitar

This controller consists of a guitar with a vertical orientation sensor and whammy bar.

 Note
In addition to the standard control set, the standard and alternate guitar controllers implement additional vendor-specific effects.

Game Function	Gamepad	Guitar (alternate)
Game Dependent	Right Trigger	Accelerometer
Game Dependent	Left Trigger	Accelerometer
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)	Strum Up, Strum Down, DPad Left, DPad Right
Game Dependent	A, B, X, Y button	Green (1st Neck Button), Red (2nd Neck Button), Blue (4th Neck Button), Yellow (3rd Neck Button)
Game Dependent	Left Bumper	Orange (5th Neck Button)
Not used	Right Bumper	Optional Guitar Effect 3

Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens Xbox 360 Console's on screen menu. It wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Not used	Left Stick X-axis	Not used
Not used	Left Stick Y-axis	Not used
Game Dependent	Right Stick X-axis	Whammy Bar. When the whammy bar is held out from the base of the controller, it is in an Out (or neutral) position and returns a full negative value (-32,768). When the whammy bar is held against the base of the controller, it is in an In position and registers a full positive value (32,767).
Game Dependent	Right Stick Y-axis	Vertical Accelerometer When the controller is held at an angle of +55° or more, the Y-axis of the right stick registers a full positive max value. Values less than 55° register a positive value between 0 and 32,767. When the controller is held at an angle of -55° or more, the X-axis of the right stick registers a full negative max value. Values less than -55° register a negative value between 0 and -32,768.
Not used	Left and Right Stick buttons	Not used

Arcade Stick

This controller consists of a digital joystick and buttons, which are commonly used for arcade and fighting games.

Game Function	Gamepad	Arcade Stick
Game Dependent	Right Trigger	RT button (optional)
Game Dependent	Left Trigger	LT button (optional)
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)	Arcade Stick (Up/Down/Left/Right)
Game Dependent	A, B, X, Y button	Green/Red/Blue/Yellow button
Game Dependent	Left Bumper	Optional
Game Dependent	Right Bumper	Optional
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens the on-screen Xbox 360 menu. It wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Not used	Left and Right Stick X-axis	Not used
Not used	Left and Right Stick Y-axis	Not used

Big Button Pad

This controller consists of one big button and several other buttons.

Game Function	Gamepad	Big Button Pad
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)	Directional Pad (Up/Down/Left/Right)

Game Dependent	A, B, X, Y button	A, B, X, Y button
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens the on-screen Xbox 360 menu. It wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Game Dependent	Not applicable	Big Button

Dance Pad

This controller consists of a floor mat, which is used exclusively with dance games. An optional headset extension cable might be bundled with the dance pad.

Game Function	Gamepad	Dance Pad
Not used	Right Trigger	Not used
Not used	Left Trigger	Not used
Game dependent	Directional Pad (Up/Down/Left/Right)	Top/North, Bottom/South, West/Left, East/Right pads
Game dependent	A, B, X, Y button	A Pad (North East), B Pad (North West), X Pad (South East), Y Pad (South West)
Not used	Left Bumper	Not used
Not used	Right Bumper	Not used
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button/Pad
Back	Back button	Back button/Pad
Opens the on-screen Xbox 360 menu. Wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button/Pad
Not used	Left Stick X and Y-axes	Not used
Not used	Right Stick X and Y-axes	Not used
Not used	Left and Right Stick buttons	Not used

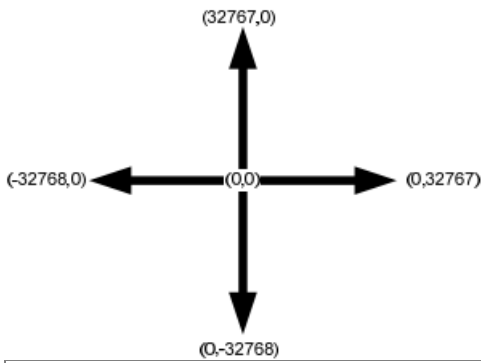
Drum Kit

This controller consists of a drum kit, which is used for music games. Bass drum control is implemented as a foot pedal with various drums represented by four (core kit) or seven (extended drum kit) pads.

Game Function	Gamepad	Drum Kit
Not used	Right Trigger	Not used
Not used	Left Trigger	Not used
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)	Directional Pad
Game Dependent	A, B, X, Y button	Green, Red, Blue, and Yellow pads.
Game Dependent	Left Bumper	Orange pedal
Game Dependent	Right Bumper	Extra Pad 3 (optional)
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens the on-screen Xbox 360 menu. Wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Game Dependent	Left Stick button	Extra Pad 2 (optional)
Game Dependent	Right Stick button	Extra Pad 1 (optional)

Flight Stick

This controller consists of an aircraft joystick, which is used for flight simulation, air combat, and space combat games. The joystick's position is mapped to the Left Stick's X and Y-axes as on a Cartesian plane.



Game Function	Gamepad	Flight Stick
Throttle (Up/Down)	Right Trigger	Throttle Control (Forward/Backward)
Rudder (Left/Right)	Left Trigger	Twisting handle or rocker switch
Menu Navigation	Directional Pad (Up/Down/Left/Right)	Hat switch
Primary weapon, Secondary weapon, Action, Action	A, B, X, Y button	Joystick trigger, B, X, Y buttons
Optional	Left Bumper	Optional
Optional	Right Bumper	Optional
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens the on-screen Xbox 360 menu. Wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Roll (Left/Right)	Left Stick X-axis	Joystick X-axis
Pitch (Up/Down)	Left Stick Y-axis	Joystick Y-axis
Point Of View	Right Stick X and Y-axis	Right Stick X and Y-axis
Optional	Left and Right Stick buttons	Optional

Gamepad

This is the standard Xbox 360 controller. It is suitable for emulating all other controller subtypes.

Game Function	Gamepad
Game Dependent	Right Trigger
Game Dependent	Left Trigger
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)
Game Dependent	A, B, X, Y button
Game Dependent	Left Bumper
Game Dependent	Right Bumper
Pause/Wakes controller from standby and then signals console to wake up.	Start Button
Back	Back button
Opens the on-screen Xbox 360 menu. Wakes controller from standby and then signals console to wake up.	Xbox Guide Button
Not used	Left and Right Stick X-axis
Not used	Left and Right Stick Y-axis

Guitar

This controller consists of a guitar with a vertical orientation sensor and whammy bar. In addition, some controllers implement vendor-specific optional effects.

Game Function	Gamepad	Guitar

Game Dependent	Right Trigger	Optional Guitar Effect 2 (Right Effect Pedal)
Game Dependent	Left Trigger	Optional Guitar Effect 1 (Right Effect Pedal)
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)	Strum Up, Strum Down, DPad Left, DPad Right
Game Dependent	A, B, X, Y button	Green (1st Neck Button), Red (2nd Neck Button), Blue (4th Neck Button), Yellow (3rd Neck Button)
Game Dependent	Left Stick button + A	Left Stick button + A
Game Dependent	Left Stick button + B	Left Stick button + B
Game Dependent	Left Stick button + X	Left Stick button + X
Game Dependent	Left Stick button + Y	Left Stick button + Y
Game Dependent	Left Bumper	Orange (5th Neck Button)
Game Dependent	Lower Left Bumper	Left Stick button + Left Bumper
Not used	Right Bumper	Optional Guitar Effect 3
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens Xbox 360 Console's on screen menu. Wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Not used	Left Stick X-axis	Not used
Not used	Left Stick Y-axis	Not used
Game Dependent	Right Stick X-axis	Whammy Bar. When the whammy bar is held out from the base of the controller, it is in an Out (or neutral) position and returns a full negative value (-32,768). When the whammy bar is held against the base of the controller, it is in an In position and registers a full positive value (32,767).
Game Dependent	Right Stick Y-axis	Tilt Sensor When the controller is held at an angle of +55° or more, the Y-axis of the right stick registers a full positive max value. Values less than 55° register a positive value between 0 and 32,767. When the controller is held at an angle of -55° or more, the X-axis of the right stick registers a full negative max value. Values less than -55° register a negative value between 0 and -32,768.

Not used	Left and Right Stick buttons	Not used
----------	------------------------------	----------

Wheel

Several wheels (with associated pedals) are available for the Xbox 360 platform.

Game Function	Gamepad	Wheel
Accelerate	Right Trigger	Accelerator Pedal
Brake	Left Trigger	Brake Pedal
Game Dependent/Menu Navigation	Directional Pad (Up/Down/Left/Right)	Directional Pad (Up/Down/Left/Right)
Game Dependent	A, B, Y, X button	A, B, Y, X button
Gear Down	Left Bumper	Left Paddle/Shift Lever
Gear Up	Right Bumper	Right Paddle/Shift Lever
Pause/Wakes controller from standby and then signals console to wake up.	Start Button	Start Button
Back	Back button	Back button
Opens Xbox 360 Console's on-screen menu. Wakes controller from standby and then signals console to wake up.	Xbox Guide Button	Xbox Guide Button
Steering	Left Stick X-axis	Wheel Rotation
Not used	Left Stick Y-axis	Not used
Optional	Right Stick X-axis	Optional
Optional	Right Stick Y-axis	Optional

See Also

Conceptual

[Input](#)

[Zune Button Mapping](#)

How To: Detect Whether a Controller Button Is Pressed

Demonstrates how to detect whether a user has pressed a digital button on a connected Xbox 360 Controller.

The topic shows you how to check whether the digital button currently is held in the down position. For every update loop in which the **A** button remains pressed, the controller will increase vibration. To detect only the first time a button has been pressed and ignore when the button is continuously held down, such as when you want to test how fast a player can rapidly press a button, see [How To: Detect Whether a Controller Button Has Been Pressed This Frame](#).

Tip

This technique is mentioned in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: What's My Motivation](#).

Detecting the Current Position of the Controller Button

To detect whether a controller button is currently being pressed

1. Get the state of the Xbox 360 Controller by using [GetState](#).
2. Verify that the controller currently is connected by retrieving the [IsConnected](#) property.
3. Retrieve the values of the [Buttons](#) you wish to check.

If the current state is **Pressed**, the button is being pressed currently.

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace InputDetectSimple
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();
        }

        protected override void LoadContent()
        {
        }

        protected override void UnloadContent()
        {
        }

        protected override void Update(GameTime gameTime)
        {
            // Allow the game to exit.
        }
    }
}
```

```

        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
            this.Exit();

        UpdateInput();

        base.Update(gameTime);
    }

    float vibrationAmount = 0.0f;

    void UpdateInput()
    {
        // Get the current gamepad state.
        GamePadState currentState = GamePad.GetState(PlayerIndex.One);

        // Process input only if connected and button A is pressed.
        if (currentState.IsConnected && currentState.Buttons.A ==
            ButtonState.Pressed)
        {
            // Button A is currently being pressed; add vibration.
            vibrationAmount =
                MathHelper.Clamp(vibrationAmount + 0.03f, 0.0f, 1.0f);
            GamePad.SetVibration(PlayerIndex.One,
                vibrationAmount, vibrationAmount);
        }
        else
        {
            // Button A is not being pressed; subtract some vibration.
            vibrationAmount =
                MathHelper.Clamp(vibrationAmount - 0.05f, 0.0f, 1.0f);
            GamePad.SetVibration(PlayerIndex.One,
                vibrationAmount, vibrationAmount);
        }
    }

    protected override void Draw(GameTime gameTime)
    {
        graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
        base.Draw(gameTime);
    }
}

```

See Also [How To: Detect Whether a Controller Button Has Been Pressed This Frame](#)
[How To: Detect Whether a Controller Is Disconnected](#)

How To: Detect Whether a Controller Button Has Been Pressed This Frame

Demonstrates how to detect whether a user has just pressed a digital button on a connected Xbox 360 Controller. This topic ignores held buttons. It only activates when a button has been pressed in the current frame.

The example uses a stored [GamePadState](#) object and compares the value of the **A** button to determine whether the state of the button has changed from **Released** to **Pressed**. If it has changed, it indicates the user pressed the button.

Unlike the more basic example shown in [How To: Detect Whether a Controller Button Is Pressed](#), this topic only reports a digital button press, not when the button is simply held down. This can be useful in games where you want to test the user's ability to rapidly press a button. In a manner similar to [How To: Detect Whether a Controller Button Is Pressed](#), the result of this sample is vibration. However, the only way to increase the vibration amount is to rapidly press the **A** button. Holding the button will not increase the vibration.

Tip

This technique is implemented in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: What's My Motivation](#).

Detecting Whether a Controller Button Has Been Pressed This Frame

To detect whether a controller button has just been pressed this frame

1. Get the state of the Xbox 360 Controller by using [GetState](#).
2. Verify that the controller is currently connected by retrieving the [IsConnected](#) property.
3. Compare the values of the [Buttons](#) you want to check between the current state and previous state.

If the current state is **Pressed** and the previous state is **Released**, the button has been pressed.

4. Update the previous state to the new state.

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace InputDetect
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();
            previousGamePadState = GamePad.GetState(PlayerIndex.One);
        }

        protected override void LoadContent()
        {
        }
    }
}
```

```

    {
    }

    protected override void UnloadContent()
    {
    }

    protected override void Update(GameTime gameTime)
    {
        // Allow the game to exit.
        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
            this.Exit();

        UpdateInput();

        base.Update(gameTime);
    }

    GamePadState previousGamePadState;
    float vibrationAmount = 0.0f;

    void UpdateInput()
    {
        // Get the current gamepad state.
        GamePadState currentState = GamePad.GetState(PlayerIndex.One);
        // Process input only if connected.
        if (currentState.IsConnected)
        {
            // Increase vibration if the player is tapping the A button.
            // Subtract vibration otherwise, even if the player holds down A
            if (currentState.Buttons.A == ButtonState.Pressed &&
                previousGamePadState.Buttons.A == ButtonState.Released)
            {
                // Button A has just been pressed; add vibration.
                vibrationAmount =
                    MathHelper.Clamp(vibrationAmount + 0.3f, 0.0f, 1.0f);
                GamePad.SetVibration(PlayerIndex.One,
                    vibrationAmount, vibrationAmount);
            }
            else
            {
                // Subtract some vibration.
                vibrationAmount =
                    MathHelper.Clamp(vibrationAmount - 0.04f, 0.0f, 1.0f);
                GamePad.SetVibration(PlayerIndex.One,
                    vibrationAmount, vibrationAmount);
            }

            // Update previous gamepad state.
            previousGamePadState = currentState;
        }
    }

    protected override void Draw(GameTime gameTime)
    {
        graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
        base.Draw(gameTime);
    }
}

```

See Also [How To: Detect Whether a Controller Button Is Pressed](#)
[How To: Detect Whether a Controller Is Disconnected](#)

How To: Detect Whether a Controller Is Disconnected

Demonstrates how to detect whether an Xbox 360 Controller connected to the system at game start has become disconnected.

This topic shows you how to check the `IsConnected` property of a retrieved `GamePadState` object. If the value is **true**, the controller is still connected, and the game continues. If the value is **false**, the controller has become disconnected, and the game will pause until the user reconnects the controller.

Detecting Whether a Controller Has Been Disconnected

To detect whether a controller has been disconnected

1. At program start, present an interface for users to choose which controllers are connected, and store this information.
2. Proceed to the game loop.
3. In the game loop, get the state of each Xbox 360 Controller by using `GetState`.
4. Retrieve the `IsConnected` property to verify that each controller from the earlier list of connected controllers is still connected.

If the current value of `IsConnected` is **false** for a previously-connected controller, that controller has become disconnected.

5. Enter a pause loop and wait for the user to reconnect the controller.

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace InputDisconnect
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        // Game states.
        enum GameState
        {
            InputSetup = 0,
            Game,
            Disconnected
        }

        // Current state of the game.
        GameState gameState = GameState.InputSetup;

        // Controllers that are in use.
        bool[] activeControllers = new bool[4];

        // Disconnected controller detected.
        bool disconnectDetected = false;

        // Background color. Use this to indicate a disconnect.
        Color backColor = Color.CornflowerBlue;

        public Game1()
```

```

{
    graphics = new GraphicsDeviceManager(this);
}

protected override void Initialize()
{
    base.Initialize();
}

protected override void LoadContent()
{
}

protected override void UnloadContent()
{
}

protected override void Update(GameTime gameTime)
{
    // Allow the game to exit.
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    UpdateInput();

    switch (gameState)
    {
        case GameState.InputSetup:
            // Allow players to join the game,
            // and determine active controllers.
            // In this example, there is only one player.
            activeControllers[0] = true;

            // When ready, proceed to the game.
            gameState = GameState.Game;
            break;

        case GameState.Game:
            // If disconnected, go to the disconnect loop.
            if (disconnectDetected)
            {
                backColor = Color.Black;
                gameState = GameState.Disconnected;
            }
            break;

        case GameState.Disconnected:
            // If reconnected, continue to the game.
            if (!disconnectDetected)
            {
                backColor = Color.CornflowerBlue;
                gameState = GameState.Game;
            }
            // Otherwise, pause the game and display a message.
            break;
    }

    base.Update(gameTime);
}

void UpdateInput()
{
    disconnectDetected = false;
    PlayerIndex index = PlayerIndex.One;
    for (int i = 0; i < 4; i++, index++)
    {
        if (activeControllers[i] &&

```

```
        !GamePad.GetState(index).IsConnected)
    {
        disconnectDetected = true;
    }
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(backColor);
    base.Draw(gameTime);
}
}
```

See Also [How To: Detect Whether a Controller Button Is Pressed](#)

How To: Get the Current Mouse Position (Windows)

Demonstrates how to get the current horizontal and vertical position of the mouse cursor relative to the upper-left corner a game window.

Note

This example applies only to Windows development. The [Mouse](#) and [MouseState](#) objects are not supported on Xbox 360.

The example queries the [Mouse](#) class to return a [MouseState](#) object that holds the button states and the current position of the mouse relative to the upper-left corner of the game window.

Retrieving the Current Mouse Position

To retrieve the current mouse position

1. Call [GetState](#) to get the current state of the mouse in a [MouseState](#) object.
2. Access the [X](#) and [Y](#) properties of the [MouseState](#) retrieved in the previous section to get the X (horizontal) and Y (vertical) mouse position, in pixels, relative to the upper-left corner of the game window.

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace MousePosition
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        Color backColor = Color.CornflowerBlue;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();

            // Set this to true to make the mouse cursor visible.
            // Use the default (false) if you are drawing your own
            // cursor or don't want a cursor.
            this.IsMouseVisible = true;
        }

        protected override void LoadContent()
        {
        }

        protected override void UnloadContent()
        {
        }

        protected override void Update(GameTime gameTime)
        {
        }
    }
}
```

```

    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    UpdateMouse();

    base.Update(gameTime);
}

protected void UpdateMouse()
{
    MouseState current_mouse = Mouse.GetState();

    // The mouse x and y positions are returned relative to the
    // upper-left corner of the game window.
    int mouseX = current_mouse.X;
    int mouseY = current_mouse.Y;

    // Change background color based on mouse position.
    backColor = new Color((byte)(mouseX / 3), (byte)(mouseY / 2), 0);
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(backColor);
    base.Draw(gameTime);
}
}
}

```

See Also

How To: Detect Whether a Key Is Pressed

Demonstrates how to detect whether a user has pressed or released a key on the keyboard.

By using [GetState](#), a game can determine which keys are being held down. Often a game needs to detect when a user has pressed or released a button. For example, there is the case of an action title that requires users to press and release keys in rapid succession. The example uses a cached [KeyboardState](#) object to determine whether keys have been pressed or released in a given frame.

Depending on game design, there may be times when checking for a key press needs to occur more frequently, and other times when the check may not be needed as often. It is possible in the case of very fast key presses that more than one key press could occur within one frame. In this case, the last key press will be returned. Writing code that checks for key presses as often as possible when needed is the best way to handle this case.

Tip

This technique is implemented in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: What's My Motivation](#).

Detecting Whether a Key is Pressed or Released

To detect whether a key is pressed or released

1. Declare a [KeyboardState](#) object to hold the last known keyboard state (in this example, the **oldState** object).
2. Assign this object a value in your constructor.
3. Call [GetState](#) to retrieve the current keyboard state (in this example, the **newState** object).
4. Compare the values in your **oldState** object to the values in the **newState** object.

Keys pressed in the **newState** object that were not pressed in the **oldState** object have been pressed during this frame. Conversely, keys pressed in the **oldState** object that are not pressed in the **newState** object have been released during this frame.

5. Update **oldState** object to the **newState** object before leaving **Update**.

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace Keypress
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        KeyboardState oldState;
        Color backColor = Color.CornflowerBlue;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();
        }
    }
}
```

```

    oldState = Keyboard.GetState();
}

protected override void LoadContent()
{
}

protected override void UnloadContent()
{
}

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    UpdateInput();

    base.Update(gameTime);
}

private void UpdateInput()
{
    KeyboardState newState = Keyboard.GetState();

    // Is the SPACE key down?
    if (newState.IsKeyDown(Keys.Space))
    {
        // If not down last update, key has just been pressed.
        if (!oldState.IsKeyDown(Keys.Space))
        {
            backColor =
                new Color(backColor.R, backColor.G, (byte)~backColor.B);
        }
    }
    else if (oldState.IsKeyDown(Keys.Space))
    {
        // Key was down last update, but not down now, so
        // it has just been released.
    }

    // Update saved state.
    oldState = newState;
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(backColor);
    base.Draw(gameTime);
}
}
}

```

Audio

Provides classes and methods for playing audio files.

In This Section

[Audio Overview](#)

The XNA Framework provides audio playback through the Microsoft Cross-Platform Audio Creation Tool (XACT), and a simple [SoundEffect](#) class for audio playback that doesn't require XACT.

[Attenuation and Doppler Pitch Shifting Overview](#)

Provides an overview of attenuation and Doppler pitch shifting.

[How To: Play a Sound](#)

Demonstrates how to play a sound.

[How To: Play a Song](#)

Demonstrates how to play a song from a user's media library.

[How To: Loop a Sound](#)

Demonstrates how to loop a sound.

[How To: Change the Pitch or Volume of a Sound](#)

Demonstrates how to change pitch and volume of a playing sound.

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Demonstrates how to apply 3D positioning effects to SoundEffects.

[How To: Add a Sound File to Your Game Using XACT](#)

Demonstrates how to add wave (.WAV) files to an XACT project that can be built and interpreted by an XNA Game Studio game to play audio.

[How To: Play a Sound Using XACT](#)

Demonstrates how to initialize the audio engine, load sound and wave banks, and play a sound by using a [Cue](#) object.

[How To: Specify Background Music Using XACT](#)

Demonstrates how to specify sound (.WAV) files as background music.

[How To: Stop or Pause a Sound Using XACT](#)

Demonstrates how to initialize the audio engine; load a sound bank and wave bank; and play, pause, resume, or stop a sound (called a cue).

[How To: Change Sound Volume Levels Using XACT](#)

Demonstrates how to initialize the audio engine and how to use categories to change the playback volume of a group of sounds.

[How To: Stream a Sound Using XACT](#)

Demonstrates how to create and use a streaming wave bank.

[How To: Apply Basic 3D Positional Effects to a Cue](#)

Demonstrates how to apply 3D positioning effects to cues.

[How To: Apply Attenuation and Doppler 3D Audio Effects](#)

Demonstrates how to apply attenuation and Doppler 3D positioning effects in code.

See Also

Concepts

[Audio Content Catalog at XNA Creators Club Online](#)

Audio Overview

The XNA Framework provides audio playback through the Microsoft Cross-Platform Audio Creation Tool (XACT), and a simple [SoundEffect](#) class for audio playback that doesn't require XACT.

Audio Orientation

Some general terminology is provided below, followed by a more detailed discussion.

Wave

A wave is an audio data file used independently, or as a building block for creating game sound effects.

Wave Bank

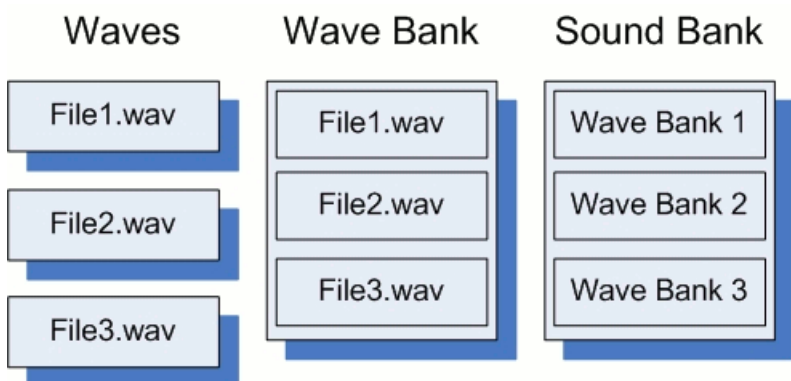
A wave bank is multiple wave files logically grouped into a single file.

Sound Bank

A sound bank is a logically grouped collection of sounds (wave banks) and cues.

Cue

A cue allows a game programmer to play sounds. It is composed of one or more sounds, and is referenced through a sound bank.



Simple Audio Playback

The simplest way to play sounds for background music or sound effects is to use [SoundEffect](#) and [SoundEffectInstance](#). Source audio files are added to the project like any other game assets. For example code, see [How To: Play a Sound](#), [How To: Loop a Sound](#), and [How To: Change the Pitch or Volume of a Sound](#). For background music, see [How To: Play a Song](#).

[SoundEffect](#) and [SoundEffectInstance](#) are available on Windows, the Xbox 360, and Zune. However, these APIs are the only audio option on Zune.

Introducing XACT

XACT is at the heart of the XNA Framework Audio API. XACT combines two things. It combines a powerful graphical tool for authoring audio content with an API that is responsible for interpreting the authored XACT files and playing audio in response to game events. It is expected that core audio elements (waves) will be fully developed before using XACT.

Getting Started with XACT

Not all features of XACT are necessary for a game title that uses simple audio playback. To learn how to create a simple XACT project that contains only a few wave files, see [How To: Add a Sound File to Your Game Using XACT](#). To learn how to load and play cues from the XACT project you have built, see [How To: Play a Sound Using XACT](#). To create simple sound effects without using XACT at all, see [How To: Play a Sound](#).

What XACT Does

As a GUI-driven audio content creation system, XACT enables audio designers to load wave files into groups, organize the files into discrete *cues* that can be activated by in-game events, and create transitions between cues. XACT also enables designers to define variables that can be changed in-game to modify audio settings. With these advanced tools, an audio designer might, for example, design a set of car engine sounds for a racing game, and through the use of a variable, cause the car engine sounds to increase or decrease in pitch and volume as the variable is controlled in-game by the XACT engine.

To get started using XACT

1. Click the **Start** menu, and then click **All Programs**.
2. Click the **XNA Game Studio** folder, then **Tools**, and then click **Microsoft Cross-Platform Audio Creation Tool (XACT)**.

For detailed information about how to author audio in the XACT tool, including information about categories, variables, and other advanced features, see [XACT Audio Authoring](#).

Programming for XACT

Once you create an XACT project and save it as an .xap file, add the .xap file and any wave files the XACT project uses as input to your XNA Game Studio game. The Content Pipeline will build the needed files for you to access your content at run time.

An XACT project builds a set of files: a global settings file (.xgs), one or more wave banks (.xwb), and one or more sound banks (.xsb). These files may be provided to [AudioEngine](#), [WaveBank](#), and [SoundBank](#) constructors, respectively.

To initialize the XACT engine, you must create a new [AudioEngine](#), and provide the path to the global settings file. Then, load any wave banks you need by creating new [WaveBank](#) objects, and load any sound banks that you need by creating [SoundBank](#) objects. Once you load the necessary files, you can access cues created by the audio designer by calling [GetCue](#) on the [SoundBank](#) that contains the [Cue](#) that you want to retrieve. Each [Cue](#) instance that you retrieve is unique, even when you retrieve multiple cues with the same name. This allows multiple instances of the same [Cue](#) to exist and play simultaneously.

You can play, pause, resume, and stop [Cue](#) objects by using the [Play](#), [Pause](#), [Resume](#), and [Stop](#) methods, respectively. See [How To: Play a Sound Using XACT](#) for information about how to play a cue. For more information about how to pause, resume, and stop cues, see [How To: Stop or Pause a Sound Using XACT](#).

Periodically, you must call [Update](#) to allow the audio engine to process audio data.

For more advanced projects, you can also access [AudioCategory](#) objects for controlling the playback of sound categories by calling [AudioEngine.GetCategory](#). Also, to access variables that the audio designer has tied to playback changes in volume, pitch, or DSP effects, you can call [AudioEngine.GetGlobalVariable](#). For more information about how to use categories to change sound volume levels, see [How To: Change Sound Volume Levels Using XACT](#).

3D Audio

Both the [Cue](#) and [SoundEffect](#) classes provide the ability to place audio in a 3D space. By creating an [AudioEmitter](#) and [AudioListener](#) object, both sound playback APIs can position a sound in 3D, and change the 3D position of a sound during playback. Once you create and initialize the [AudioEmitter](#) and [AudioListener](#), call [Cue.Apply3D](#) or [SoundEffectInstance.Apply3D](#).

See Also

Concepts

[Audio Content Catalog at XNA Creators Club Online](#)

Attenuation and Doppler Pitch Shifting Overview

Provides an overview of attenuation and Doppler pitch shifting.

Attenuation

In 3D audio, *attenuation* refers to volume attenuation—the effect of having volume fade with distance between emitter and listener. By default, XACT does not automatically apply volume attenuation as a result of calculating distance, or Doppler effect as a result of relative emitter and listener velocity.

The programmer controls the position and velocity of game objects, and the sound designer determines how that affects sound. Because XACT is designed to partition and simplify the work of the programmer and sound designer, XACT determines emitted sound by combining effects specified by the sound designer with positional information supplied by the game.

XACT exposes variables as placeholders for runtime game information such as position and velocity. The sound designer uses XACT RPCs to define the relationships between those variables and sound effects. The variables are then set at run time, either by the game or by XACT itself, and produce the desired sounds.

The project tree region of the XACT tool shows the available variables:

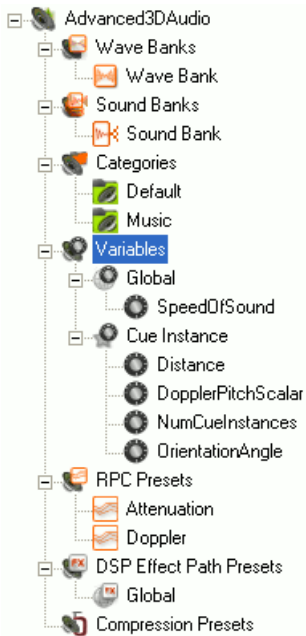


Figure 1. XACT Variables

Global variables are shared by all active sound cues. Only one of each global variable exists, and it is persisted the entire time the game is running. In contrast, a separate copy of each cue instance variable exists for every active cue. Cue instance variables persist only for the lifetime of an individual cue instance.

Global variables allow multiple audio elements to be controlled together. Cue instance variables provide individual control or information for each instance of a cue, and for each copy of the same cue.

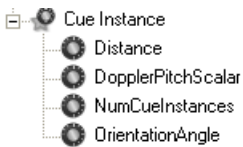


Figure 2. Cue Instance Variables

The variables used for 3D audio effects are the cue instance variables *Distance*, *DopplerPitchScalar*, and *OrientationAngle*. XACT updates these variables automatically for the cue when the game calls [Apply3D](#).

The variables provide the following information.

Variable	Description
<i>Distance</i>	The distance between the emitter and listener.
<i>DopplerPitchScalar</i>	A value calculated by comparing the relative velocities of the emitter and listener; used for Doppler pitch shifting.

<i>OrientationAngle</i>	The angle difference between the emitter and listener.
-------------------------	--

XACT updates the cue's *Distance* variable. However, by default, it does not adjust the volume for that cue. It isn't the developer's job to try to determine and set the appropriate volume. Instead, the sound designer defines the attenuation, which is the relationship between distance and volume. To do so, the designer uses the XACT tool to create an RPC. An RPC defines the mapping between an input variable, such as *Distance*, and an output parameter, such as sound volume.

The programmer might be tempted to query distance with `Cue.GetVariable("Distance")`, and then, based on distance, try to set the volume of the cue. However, volume can normally be set only for categories of cues, defined by the sound designer. See [How To: Change Sound Volume Levels Using XACT](#). The game code would also have to calculate the proper volume change based on distance. To avoid these issues, simply use an RPC as described in this topic and let XACT perform attenuation automatically, as designed.

Defining the Distance Scale

To define the Distance scale

1. Click once on the **Distance** variable in the tree.

The properties pane, below the project tree, will display properties for the *Distance* variable. Look for the section named **General**.

2. Inside the **General** section, find **Variable Range**.

The left box is the minimum value, and the right box is the maximum value.

3. Enter values into these boxes that are appropriate to the game's world scale (or whatever scale is used when calling [Apply3D](#)).

Distances outside the specified range are clamped to the range. The designer need not worry about extreme values when determining the curve. For example, a game uses a 1 m (one meter) game scale, and sounds trail off completely near 1000 m. The minimum value would be 0, and the maximum value would be 1000.

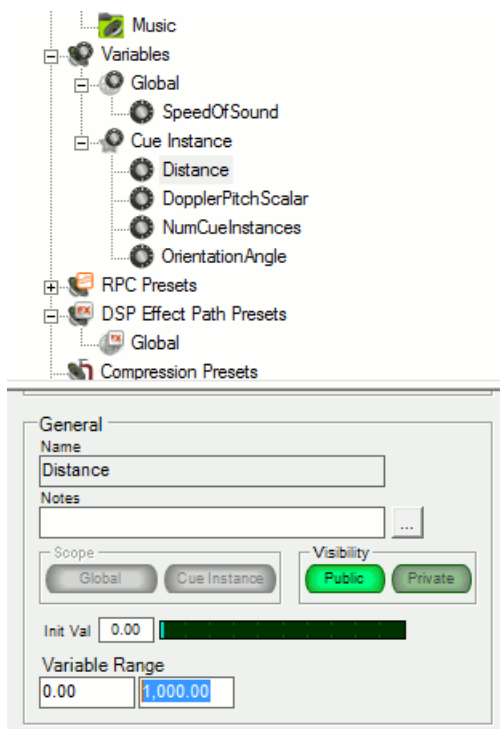


Figure 3. The Variable Range

Creating a Distance::Volume RPC

To create a Distance::Volume RPC

1. Right-click the **RPC Presets** node in the project tree, and select **New RPC Preset**.

XACT creates and displays a new RPC, and opens it for editing.

2. Set the **Variable** drop-down to **Distance** (the default is *OrientationAngle*).
3. Set the **Object** drop-down to **Sound**.

- Set the **Parameter** drop-down to **Volume**.

The RPC now adjusts volume based on distance. However, the curve is created flat, which means that volume will remain constant over distance. To change that, and define the attenuation, you must adjust the curve. In the curve plot window, distance is on the x-axis and volume, in decibels, is on the y-axis. To attenuate over distance, Y (volume) should decrease as X (distance) increases.

- To create a new control point, double-click anywhere on the plot line.

The curve is split and a new adjustable control point is added.

- Add and adjust as many curve points as needed to define the attenuation.

Different sounds may have different attenuation characteristics. These two example curves produce significantly different attenuation. The first creates a simple constant ramp from full volume (+0 dB) to silence (-96 dB) over the full distance range. The second has a fast rolloff after a short distance, then a fade out to full distance.

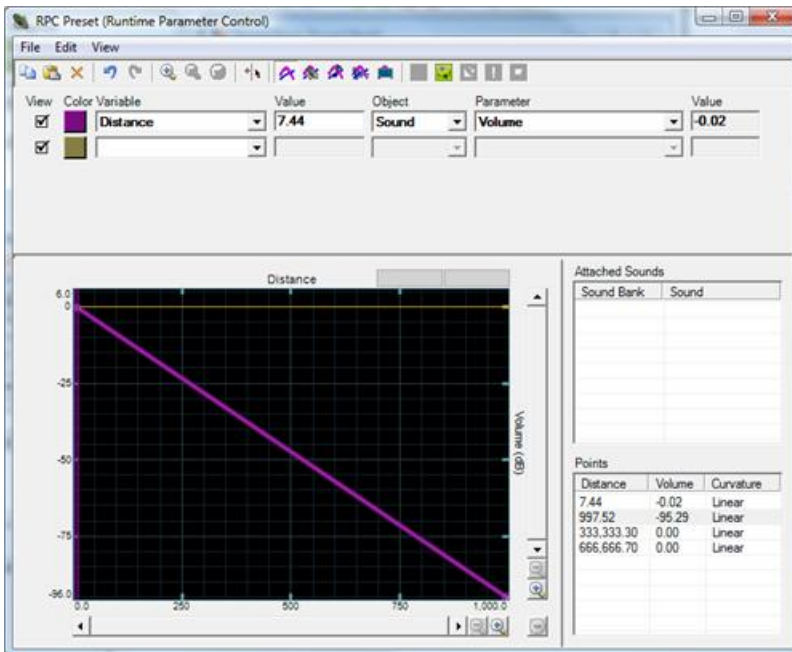


Figure 4. Attenuation RPC (Constant Rolloff)

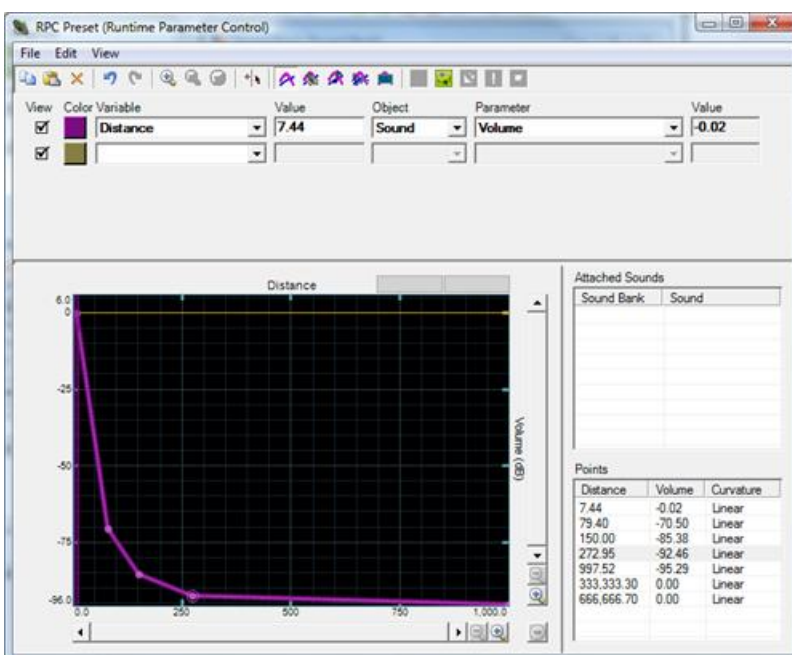


Figure 5. Attenuation RPC (Quick Rolloff)

- When finished, close the RPC window.

The new RPC is called "RPC Preset," but it can be renamed (for example, to "Attenuation") to make it easier to find in the

RPC list. The name given to the RPC does not affect the RPC or the game code.

8. To rename the RPC, right-click it and click **Rename**.

Applying the RPC to Sounds

To apply the RPC to sounds

The RPC is now defined, but no sounds are configured yet to use it. By default, sounds do not use any of the defined RPCs. Some sounds will not need attenuation (for example, narration or tutorial instructions), while others may need several different RPC effects.

1. Double-click on a sound bank listed in **Sound Banks** in the project tree.

The sound bank opens, showing the sounds and cues defined in that bank.

2. From **RPC Presets** in the project tree, drag the desired RPC onto the sound that will use that RPC.

Do this for as many sounds as will use the attenuation effect. You can apply more than one RPC to a sound. RPC curves that modify elements such as pitch and volume are additive, so if multiple parameters influence the sound, their results will be combined.

Removing an RPC from a Sound

To remove an RPC from a sound

1. To remove an RPC from a sound, either right-click the sound and click **Attach/Detach RPC(s)**, or right-click the RPC in the project tree and click **Attach/Detach Sounds**.
2. Right-click the sound and click **Attach/Detach RPC(s)**.

At this point, the attenuation RPC is defined and applied to the appropriate sounds. XACT will now use the curve to adjust emitter volume based on the distance between emitter and listener. Game code sets the [AudioEmitter.Position](#) and [AudioListener.Position](#) properties to specify the 3D location of the game objects. When the code then calls [Apply3D](#), XACT adjusts volume appropriately. See [How To: Apply Basic 3D Positional Effects to a Cue](#) for a description of the steps needed to use 3D positioning in a game.

Doppler Pitch Shifting

Doppler pitch shifting is the pitch shifting effect when a listener and an emitter are moving towards or away from each other. The relative velocities of the emitter and listener are compared and the result used to pitch up or down the emitter's sound.

Doppler pitch shifting is applied to sounds in the same way that attenuation is, except that the RPC adjusts pitch based on relative velocity rather than adjusting volume based on distance. The Doppler RPC ties the *DopplerPitchScalar* input variable to the *Sound::Pitch* output parameter.

Adding Doppler Pitch Shifting

To add doppler pitch shifting

1. Set the scale of the *DopplerPitchScalar* variable.

The default scale of 0–4 is appropriate, so leave **MinimumValue** at 0 and **MaximumValue** at 4.

The scale of *DopplerPitchScalar* is not based on the velocity scale used by the game. Instead, this value is the result of an XACT calculation to determine the pitch adjustment that should be applied due to Doppler effect. 0 indicates down 1 octave, 1 indicates unity pitch, and 4 indicates up 2 octaves.

2. To create a new RPC, right-click **RPC Presets**, and then click **New RPC Preset**.
3. Set **Variable** to **DopplerPitchScalar**, **Object** to **Sound**, and **Parameter** to **Pitch**.
4. Define the Doppler curve effect.

The y-axis is the pitch, in semitones, and ranges from down one octave (–12 semitones) to up one octave (+12 semitones).

For normal Doppler effect, the beginning of the curve would look like this:

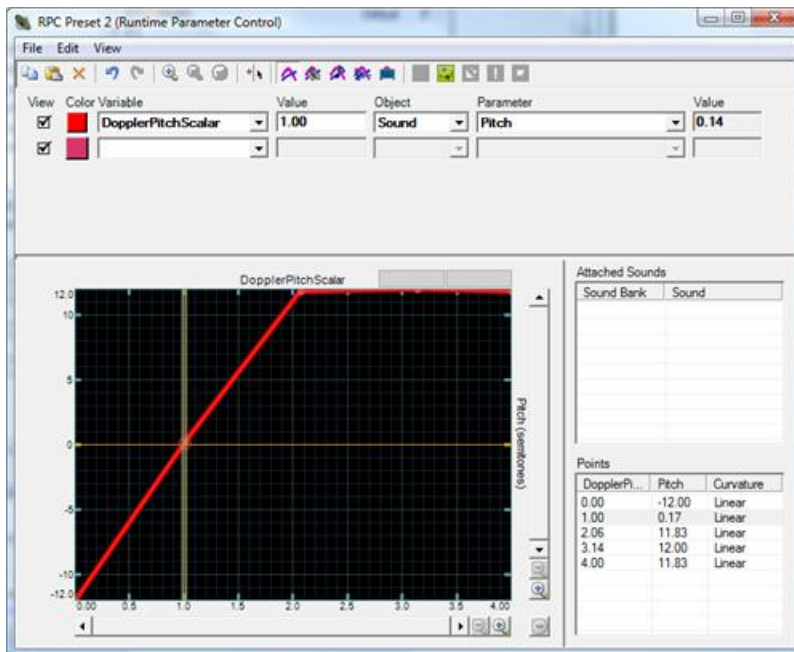


Figure 6. Doppler RPC

However, this curve limits the effect to +12 semitones. To enable a wider range effect, make the curve half as steep (+12 semitones at 4.0 rather than 2.0) and set the emitter's Doppler scale ([AudioEmitter.DopplerScale](#)) to 2.0f.

5. Close the RPC window and, if desired, rename the RPC.
6. To associate the RPC with sounds, open the sound bank, and then drag the RPC from the project tree onto the desired sound or sounds.

Code Changes to Support Doppler Pitch Shifting

XACT derives *DopplerPitchScalar* based on the relative velocities of the emitter and listener. The game informs XACT of these velocities by setting the [AudioListener.Velocity](#) and [AudioEmitter.Velocity](#) properties before calling [Apply3D](#).

The [AudioEmitter.DopplerScale](#) property can be used to exaggerate (values > 1.0) or diminish (values < 1.0) the Doppler effect.

The following example moves an object left and right. The gamepad triggers can be used to turn emitter and listener velocities on and off. Combined with an XACT project that defines RPCs as described in this topic, the code demonstrates the steps needed to enable attenuation and Doppler effects.

See Also

Concepts

[Audio Overview](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)

[How To: Apply Attenuation and Doppler 3D Audio Effects](#)

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[XACT Runtime Parameter Controls \[DirectX\]](#)

[XACT Audio Authoring \[DirectX\]](#)

Reference

[Cue.Apply3D Method](#)

[AudioEmitter.Position Property](#)

[AudioListener.Position Property](#)

[AudioEmitter.Velocity Property](#)

[AudioListener.Velocity Property](#)

[AudioEmitter.DopplerScale Property](#)

How To: Play a Sound

Demonstrates how to play a sound.

To play a sound without using XACT, use the [SoundEffect](#) or [SoundEffectInstance](#) classes. Using these classes is simpler than using XACT and the classes required by XACT.

Because XACT is not available on Zune, the [SoundEffect](#) and [SoundEffectInstance](#) will be the only option for Zune games.

[SoundEffect](#) and [SoundEffectInstance](#) are also available for Xbox and Windows XNA games.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download PlaySoundWithoutXACT_Sample.zip](#).

Adding a Sound to Your Project

To add a sound to your project

1. With your XNA Game Studio game loaded in Visual Studio, right-click the Content icon the Solution Explorer pane.
2. Click **Add**, and then click **Existing Item**.
3. Navigate to the WAV file you want to play, and then select it.

Simple Sound Playing

Declare [SoundEffect](#).

C#

```
// Audio objects
SoundEffect soundEffect;
string soundName = "kaboom";
```

Play the sound.

C#

```
ContentManager contentManager = new ContentManager(this.Services, @"Content\");
soundEffect = contentManager.Load<SoundEffect>(soundName);
soundEffect.Play();
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

Reference

[SoundEffect Class](#)

[SoundEffectInstance Class](#)

How To: Play a Song

Demonstrates how to play a song from a user's media library.

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download PlaySong_Sample.zip.](#)

The following example plays the first song from a randomly picked album.

The [Albums](#) property provides access to the media library, and the [Play](#) method will play a song. Consider any current audio playback when using the [Play](#) method. If the user has a different song playing currently, the user can use the [Stop](#) method to stop the current song.

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace PlaySong
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;

        MediaLibrary sampleMediaLibrary;
        Random rand;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";

            // Frame rate is 30 fps by default for Zune.
            TargetElapsedTime = TimeSpan.FromSeconds(1 / 30.0);

            sampleMediaLibrary = new MediaLibrary();
            rand = new Random();
        }

        protected override void Initialize()
        {
            MediaPlayer.Stop(); // stop current audio playback

            // generate a random valid index into Albums
            int i = rand.Next(0, sampleMediaLibrary.Albums.Count - 1);

            // play the first track from the album
            MediaPlayer.Play(sampleMediaLibrary.Albums[i].Songs[0]);

            base.Initialize();
        }
    }
}
```

```

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    // TODO: use this.Content to load your game content here
}

protected override void UnloadContent()
{
    // TODO: Unload any non ContentManager content here
}

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    // TODO: Add your update logic here

    base.Update(gameTime);
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    // TODO: Add your drawing code here

    base.Draw(gameTime);
}
}
}

```

See Also

Reference

[MediaPlayer Class](#)

[MediaLibrary Class](#)

[MediaPlayer.Play Method](#)

[MediaLibrary.Albums Property](#)

How To: Loop a Sound

Demonstrates how to loop a sound.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download LoopSoundWithoutXACT_Sample.zip](#).

Simple Sound Looping

1. Declare a [SoundEffect](#) using the method shown in [How To: Play a Sound](#).
2. Declare a [SoundEffectInstance](#), and set it to the return value of [SoundEffect.CreateInstance](#).
3. Set [SoundEffectInstance.IsLooped](#) to **true**.
4. Call [SoundEffectInstance.Play](#) to play the looping sound.

C#

```
SoundEffectInstance instance = soundEffect.CreateInstance();
instance.IsLooped = true;
instance.Play();
```

See Also

Tasks

[How To: Play a Sound](#)

Reference

[SoundEffect Class](#)

[SoundEffectInstance Class](#)

How To: Change the Pitch or Volume of a Sound

Demonstrates how to change pitch and volume of a playing sound.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ChangePitchAndVolumeWithoutXACT_Sample.zip](#).

The [SoundEffect.Play](#) method allows you to specify the pitch and volume of a sound for playing the sound. However, after you call [Play](#), you cannot modify the sound. Using [SoundEffectInstance](#) for a given [SoundEffect](#) allows you to change the pitch and volume of a sound at any time during playback.

Simple Sound Manipulation

1. Declare a [SoundEffect](#) using the method shown in [How To: Play a Sound](#).
2. Declare a [SoundEffectInstance](#), and set it to the return value of [SoundEffect.CreateInstance](#).

C#

```
SoundEffect soundEffect;  
SoundEffectInstance soundInstance;  
protected override void LoadContent()  
{  
    soundEffect = Content.Load<SoundEffect>("tx0_fire1");  
    soundInstance = soundEffect.CreateInstance();  
}
```

3. Call [SoundEffectInstance.Play](#) to play the sound.

C#

```
if (playerOne.Buttons.A == ButtonState.Pressed)  
{  
    // Play Sound  
    soundInstance.Play();  
}
```

4. Set [Pitch](#) and [Volume](#) to the desired values.

C#

```
// Pitch takes values from -1 to 1  
float pitch = playerOne.ThumbSticks.Left.X;  
// Volume only takes values from 0 to 1  
float volume = (playerOne.ThumbSticks.Right.X+1)/2;  
  
soundInstance.Pitch = pitch;  
soundInstance.Volume = volume;
```

See Also

Tasks

[How To: Play a Sound](#)

[How To: Loop a Sound](#)

Reference

[SoundEffect Class](#)

[SoundEffectInstance Class](#)

How To: Apply Basic 3D Positional Effects to a SoundEffect

Demonstrates how to apply 3D positioning effects to SoundEffects.

The XNA Framework audio system contains support for 3D audio positioning effects. It uses the [AudioEmitter](#) and [AudioListener](#) classes, and the [SoundEffectInstance.Apply3D](#) method. The effects simulate 3D positioning for sound by adjusting speaker mix for cues that use the 3D values.

Note that speaker mix is the only effect that will be applied automatically using this method.

The Complete Sample

The following example uses a circular rotation around a stationary [AudioListener](#) to emphasize the 3D effect.

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download Basic3DAudio_Sample.zip](#).

C#

```
using System;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;

namespace Basic3DAudioWithoutXact
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";
        }

        AudioEmitter emitter = new AudioEmitter();
        AudioListener listener = new AudioListener();
        SoundEffect soundEffect;
        SoundEffectInstance soundEffectInstance;

        protected override void LoadContent()
        {
            soundEffect = Content.Load<SoundEffect>("buzz");
            soundEffectInstance = soundEffect.CreateInstance();

            soundEffectInstance.IsLooped = true;
            soundEffectInstance.Apply3D(listener, emitter);
            soundEffectInstance.Play();
        }

        protected override void Update(GameTime gameTime)
        {
            // Allows the game to exit
            if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
                ButtonState.Pressed)
                this.Exit();

            // Move the object around in a circle.
            Vector3 objectPos = new Vector3(
                (float)Math.Cos(gameTime.TotalGameTime.TotalSeconds) / 2,
                0,
                (float)Math.Sin(gameTime.TotalGameTime.TotalSeconds));
        }
    }
}
```

```

        // Apply 3D settings to the cue.
        emitter.Position = objectPos;
        soundEffectInstance.Apply3D(listener, emitter);

        base.Update(gameTime);
    }

    protected override void Draw(GameTime gameTime)
    {
        GraphicsDevice.Clear(Color.CornflowerBlue);

        base.Draw(gameTime);
    }
}

```

Applying Basic 3D Positional Effects to a SoundEffect

To retrieve and play a SoundEffect in 3D

1. Add a WAV file to a new XNA Game Studio project as described in [How To: Play a Sound](#).
The project should contain at least one [SoundEffect](#).
2. In code, create a [SoundEffectInstance](#) for the [SoundEffect](#) when it is initially loaded.
3. Create an [AudioEmitter](#), an [AudioListener](#), and a [Vector3](#) to store the 3D position of the sound entity.
4. Load the [SoundEffect](#), and create a [SoundEffectInstance](#) to play the sound.
5. Call [Apply3D](#) on the [SoundEffectInstance](#), passing the emitter and listener.
6. Call [Play](#) to play the sound. Set the [IsLooped](#) property to true before calling [Play](#) if you want the sound to repeat.

To process audio data

1. Set the [Vector3](#) to the position from which you want the sound to come.
2. Set the [AudioEmitter.Position](#) property to this vector.
3. As an option, set the [Vector3](#) to the position where you want the listener of the 3D sound to be, and then set the [AudioListener.Position](#) property to this vector.
4. Call [SoundEffectInstance.Apply3D](#) on the cue object you retrieved previously, passing in the [AudioEmitter](#), which is an [AudioListener](#).

Note

Calling the [SoundEffectInstance.Apply3D](#) method automatically sets the speaker mix for any sound played by this sound effect to a value calculated by the difference in [Position](#) values between *listener* and *emitter*. In preparation for the mix, the sound is converted to monoaural. Any stereo information in the sound is discarded.

How To: Add a Sound File to Your Game Using XACT

Demonstrates how to add wave (.WAV) files to an XACT project that can be built and interpreted by an XNA Game Studio game to play audio.

For detailed information about how to author audio in the XACT tool, including information on categories, variables, and other advanced features, see [XACT Audio Authoring](#).

Adding a Sound File to Your Game

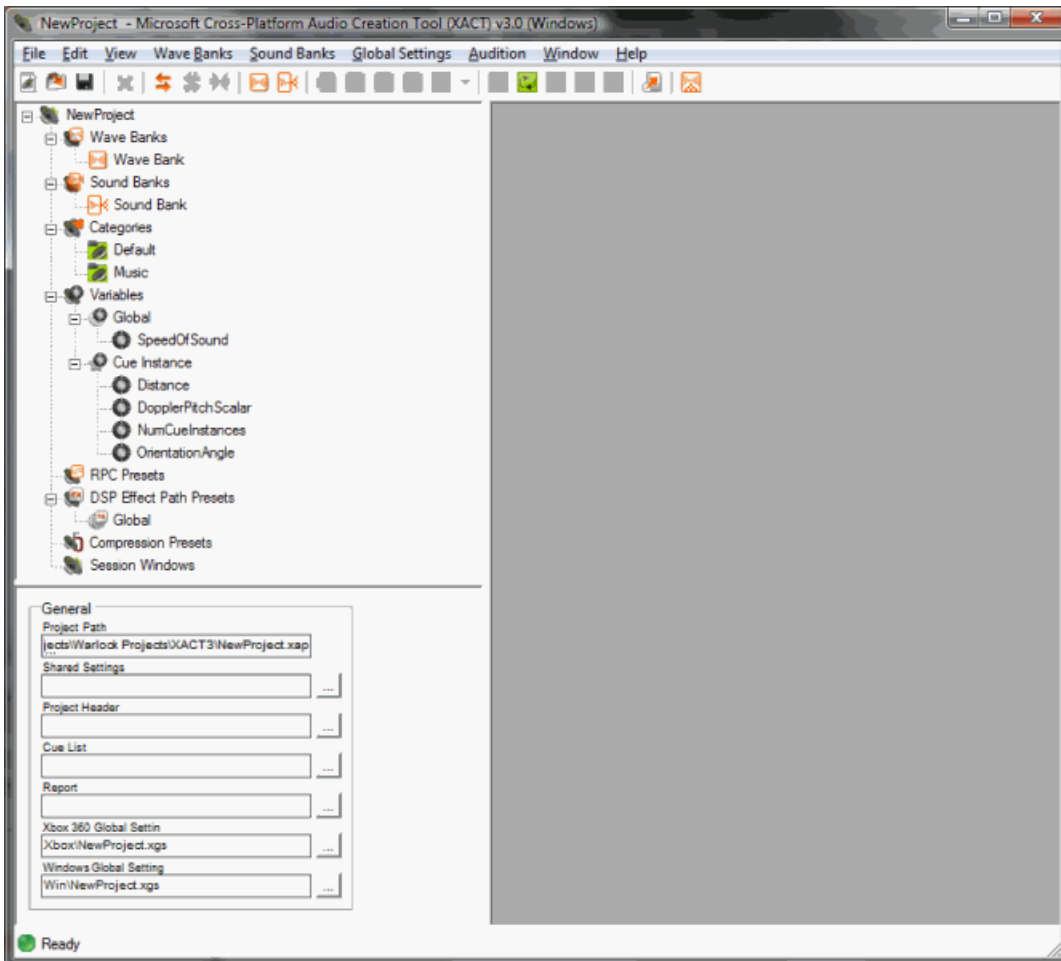
To create a wave bank and a sound bank

1. From the **Start** menu, click **All Programs**, click **XNA Game Studio 3.1**, click **Tools**, and then click **Microsoft Cross-Platform Audio Creation Tool (XACT)**.
2. To create a new project, click the **File** menu, and then click **New Project**.

You will be asked for a location and a name for your project.

3. Enter a name and a location that is easy for you to remember, and click **OK**.

The project will be created, and the screen will resemble the following:



4. In the empty project, create a wave bank. To do this, click **Wave Banks**, and then click **New Wave Bank**.

A new wave bank appears in the tree view under Wave Banks with the default name "Wave Bank."

5. Press **ENTER** to accept the default name.

6. To create a new sound bank, click **Sound Banks**, and then click **New Sound Bank**.

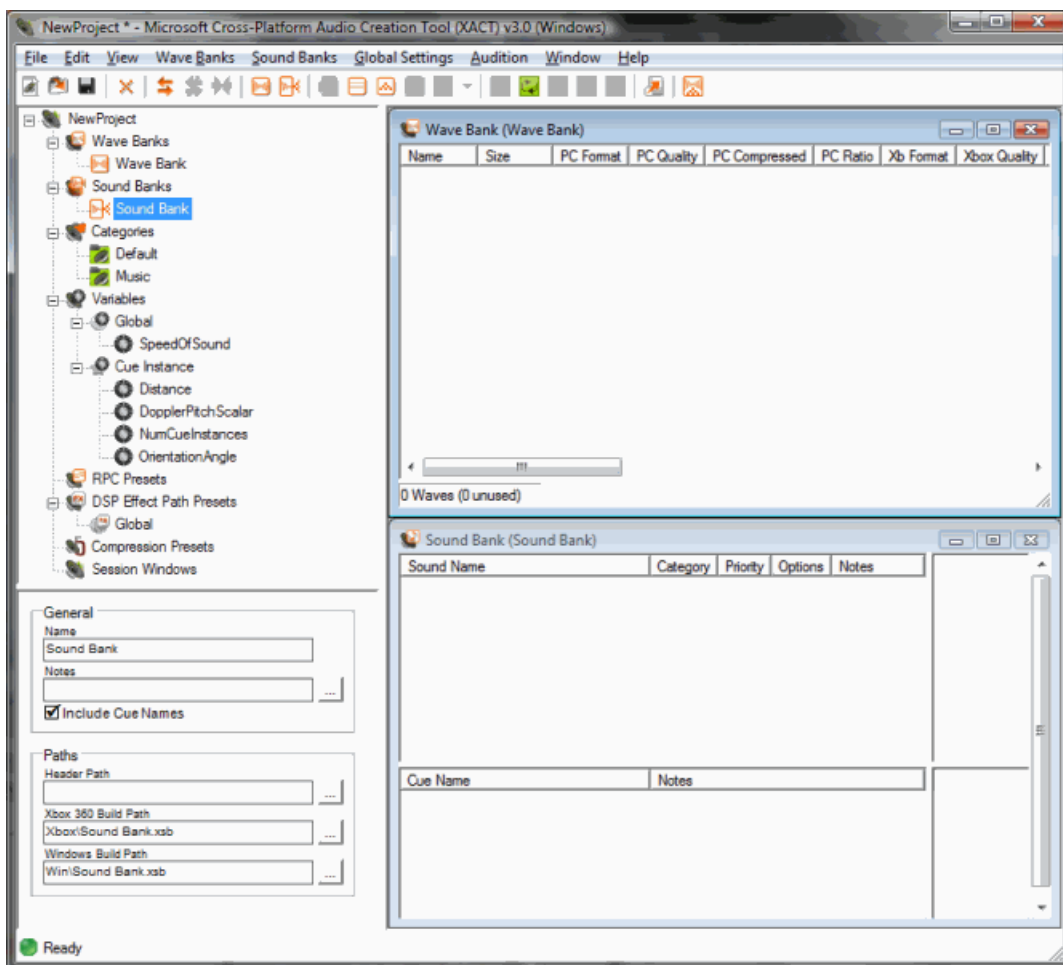
A new sound bank appears in the tree view under Sound Banks with the default name "Sound Bank."

7. Press **ENTER** to accept the default name.

At this point, two new windows have appeared: one for the wave bank and one for the sound bank.

8. To arrange these windows for easier viewing, click **Window**, and then click **Tile Horizontally**.

The windows should now look similar to the following:

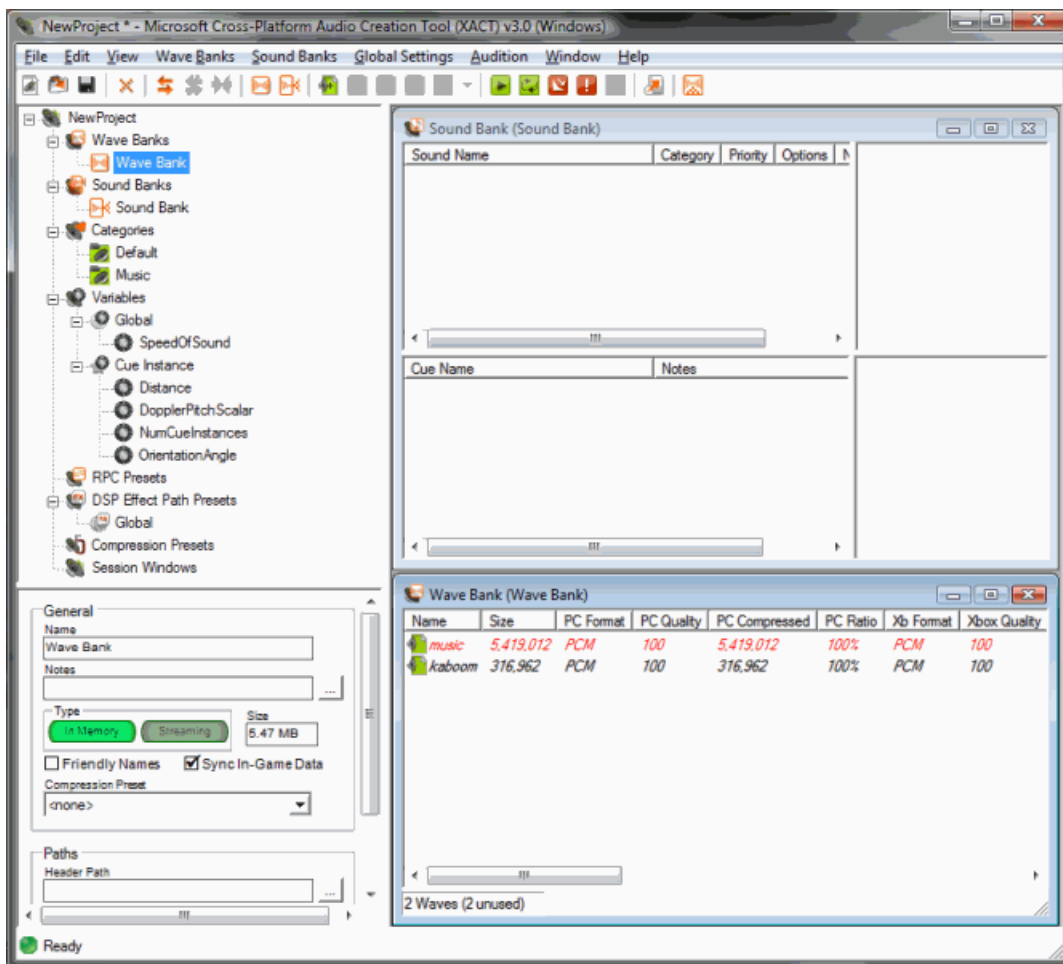


To add wave files to the wave bank window

Now you need to add your wave file to the wave bank window.

1. To make sure the wave bank window is active, click **Wave Banks**, and then click **Insert Wave File(s)**.
2. Insert one or more wave files into the wave bank.

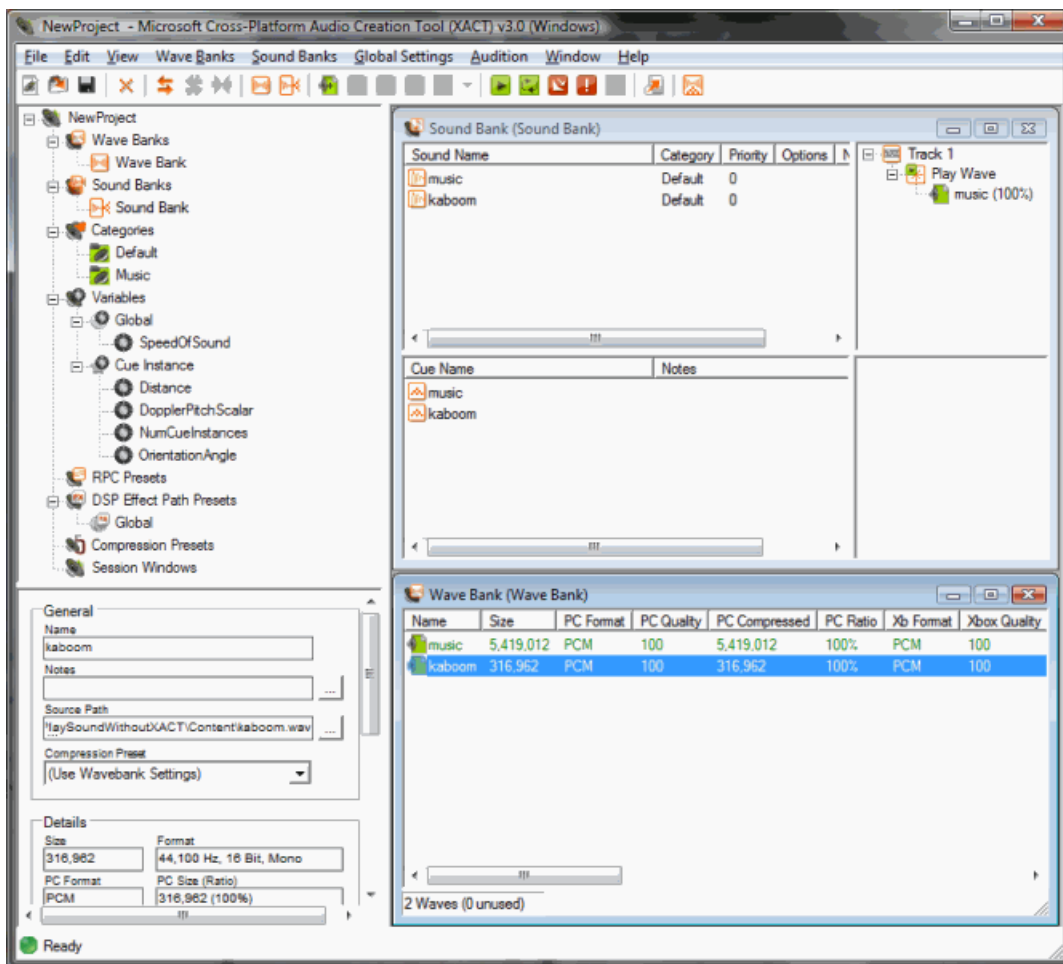
If you successfully added them, they will look similar to the following wave bank window:



Now you need to perform the following step for each wave listed in the wave bank window.

- For each wave listed in the wave bank window, drag the wave from the wave bank window to the sound bank window, and then drop the wave on top of the **Cue Name** panel.

XACT automatically creates a new cue that is linked to a new sound that plays this wave file. It should look similar to the following:



4. If the lower-left corner of the sound bank window does not have any entries, add entries to it by dragging each sound from the upper-left panel of the sound bank to the lower left panel of the sound bank.

This action creates cues that correspond to the sounds.

5. To save the project, click **File**, and then click **Save Project**.

6. Click **File**, and then **Build...** to build your project.

You are now ready to load and play sounds from this project in your game. See [How To: Play a Sound Using XACT](#) for information on how to do this.

See Also

Tasks

[How To: Play a Sound](#)

How To: Play a Sound Using XACT

Demonstrates how to initialize the audio engine, load sound and wave banks, and play a sound by using a [Cue](#) object.

Note

This example assumes you already saved an XACT project (.xap file). To learn how to do this, see [How To: Add a Sound File to Your Game Using XACT](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download PlaySound_Sample.zip](#).

C#

```
using System;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace PlaySound
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        // Audio objects
        AudioEngine engine;
        SoundBank soundBank;
        WaveBank waveBank;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();

            // Initialize audio objects.
            engine = new AudioEngine("Content\\Audio\\PlaySound.xgs");
            soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
            waveBank = new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb");

            // Play the sound.
            Cue cue = soundBank.GetCue("kaboom");
            cue.Play();
        }

        protected override void LoadContent()
        {
        }

        protected override void UnloadContent()
        {
        }

        protected override void Update(GameTime gameTime)
```



```

    {
        // Allow the game to exit.
        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
            this.Exit();

        // Update the audio engine.
        engine.Update();

        base.Update(gameTime);
    }

    protected override void Draw(GameTime gameTime)
    {
        graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
        base.Draw(gameTime);
    }
}

```

Each [Cue](#) instance that you play is unique, even when you play multiple cues with the same name. This enables you to play multiple instances of the same [Cue](#) simultaneously.

Playing a Sound

To insert the desired .xap file into your project

1. With your XNA Game Studio game loaded in Visual Studio, right-click your content icon in the Solution Explorer pane.
2. Click **Add**, click **New Folder**, and then name the new folder "Audio."
3. Move your XACT project file (.xap) and .wav file or files into the new "Audio" file created in the previous step.

The new folder will exist in the Content folder, under your game project folder.

4. Again, with your XNA Game Studio game loaded in Visual Studio, right-click the Audio file the Solution Explorer pane.
5. Click **Add**, and then click **Existing Item**.
6. Select the .xap file from the Audio folder.

The .xap file is then inserted into your project. By default, it is processed by the content pipeline, and built wave and sound banks are accessed automatically when the game is built.

7. Edit your game code.

When following these steps and using the above code, be sure your file names match those given in the example.

To play the sound

1. When your game starts, create a new [AudioEngine](#).
2. Pass in the name of the XACT global settings file output by the XACT project file.
Make sure the name includes an .xgs extension.
3. For each wave bank in your XACT project, create a new [WaveBank](#), and then pass in the audio engine and the name of the XACT wave bank file output by the XACT project file.
Make sure the name includes an .xwb extension.
4. For each sound bank in your XACT project, create a new [SoundBank](#), and then pass in the audio engine and the name of the XACT sound bank file output by the XACT project file.
Make sure the name includes an .xsb extension.
5. During game update, call the [Update](#) method of the [AudioEngine](#) to allow the audio engine to process audio data.
6. Call the [PlayCue](#) method from your [SoundBank](#), and then pass in the name of the cue you wish to play.

See Also

Tasks

[How To: Add a Sound File to Your Game Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Play a Sound](#)

How To: Specify Background Music Using XACT

Demonstrates how to specify sound (.WAV) files as background music. Specifying background music enables the XNA Framework to override the background music with the user's music choice, without interrupting other game sounds such as sound effects. This is accomplished by setting the sound bank *category* to *music* in XACT.

Note

Streaming allows for the .WAV file to be gradually read from the disk rather than loaded into memory all at once. Background music files may be larger than files containing other game sounds; therefore, typically they are streamed. In XACT, you need to create and use a streaming [WaveBank](#) to stream background music. To learn about streaming, see [How To: Stream a Sound Using XACT](#).

Specifying Background Music in Your Game

This procedure assumes you have already created an XACT project.

To specify background music in your game

1. To open the XACT project, choose the **File** menu, and then click **Open Project**.
2. Browse to your project location, and click **Open**.

The project will load.

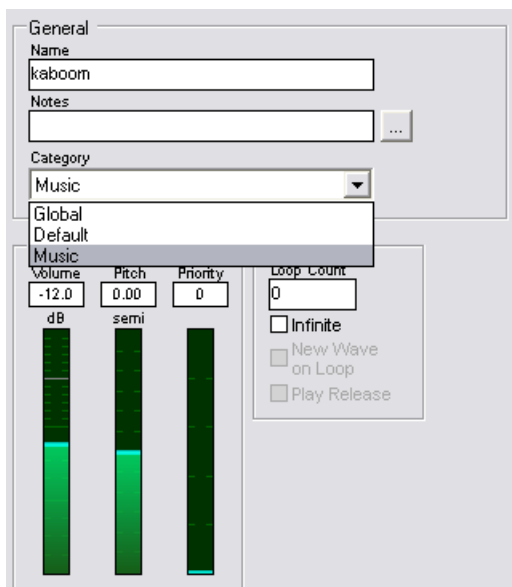
3. Double-click the sound bank that contains the background music.

This highlights the sound bank in the left navigation tree, and displays the sound bank properties pane.

4. Click the sound you wish to designate as background music.

This displays the properties pane, and then shows the category options for the sound.

5. Select **music** in the **category** drop down box, and save the project.



See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Add a Sound File to Your Game Using XACT](#)

[How To: Stream a Sound Using XACT](#)

How To: Stop or Pause a Sound Using XACT

Demonstrates how to initialize the audio engine; load a sound bank and wave bank; and play, pause, resume, or stop a sound (called a cue).

This example builds off a simpler example, [How To: Play a Sound Using XACT](#).

Note

This example assumes you already built an XACT sound bank and wave bank. To learn how to do this, see [How To: Add a Sound File to Your Game Using XACT](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample. The complete sample includes the standard solution files and any additional support files required by the sample.

[Download StopOrPauseSound_Sample.zip](#).

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace StopOrPauseSound
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        // Audio objects
        AudioEngine engine;
        SoundBank soundBank;
        WaveBank waveBank;
        Cue cue;

        GamePadState oldState;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();

            // Initialize audio objects.
            engine = new AudioEngine("Content\\Audio\\StopOrPauseSound.xgs");
            soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
            waveBank = new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb");

            // Get the cue and play it.
            cue = soundBank.GetCue("music");
            cue.Play();
        }
    }
}
```

```

protected override void LoadContent()
{
}

protected override void UnloadContent()
{
}

protected void UpdateInput()
{
    // Get the current gamepad state.
    GamePadState currentState = GamePad.GetState(PlayerIndex.One);

    if (currentState.DPad == oldState.DPad)
    {
        return;
    }

    // DPad right is 'play/pause'.
    if (currentState.DPad.Right == ButtonState.Pressed)
    {
        if (cue.IsPaused)
        {
            cue.Resume();
        }
        else if (cue.IsPlaying)
        {
            cue.Pause();
        }
        else
        {
            // If stopped, create a new cue.
            cue = soundBank.GetCue(cue.Name);
            cue.Play();
        }
    }

    // DPad left is 'stop'.
    if (currentState.DPad.Left == ButtonState.Pressed)
    {
        cue.Stop(AudioStopOptions.AsAuthored);
    }

    oldState = currentState;
}

protected override void Update(GameTime gameTime)
{
    // Allow the game to exit.
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    // Check input.
    UpdateInput();

    // Update the audio engine.
    engine.Update();

    base.Update(gameTime);
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
    base.Draw(gameTime);
}

```

```
}  
}
```

Playing, Pausing, or Stopping a Sound

To play, pause, or stop a sound

1. Create an [AudioEngine](#), [WaveBank](#), and [SoundBank](#) at game start.
2. During game update, call the [Update](#) method of the [AudioEngine](#) to enable the audio engine to process audio data.
3. Retrieve a cue you want to play by calling [SoundBank.GetCue](#), and then store the returned cue value.
4. To play a cue, pause a cue, or stop a cue, call these methods: To play the cue, call [Cue.Play](#); to pause the cue, call [Cue.Pause](#); and to stop the cue entirely, call [Cue.Stop](#).

How To: Change Sound Volume Levels Using XACT

Demonstrates how to initialize the audio engine and how to use categories to change the playback volume of a group of sounds.

Categories control the sound volume levels for groups of sounds in XACT. These categories are collections that can hold one or more references to cues. Global variables linked to runtime parameter controls (RPC) within the XACT project control the individual cue volumes. This topic explains how to use categories to control the volume of groups of sounds.

For more information about using global variables and RPCs, see [XACT Variables](#) and [XACT Runtime Parameter Controls](#). For more information on creating categories within your XACT project, see [Making XACT Categories](#). To see how XACT automatically adjusts the volume based on distance, see [How To: Apply Attenuation and Doppler 3D Audio Effects](#).

Note

This example assumes you already built an XACT sound bank and wave bank. To learn how to do this, see [How To: Add a Sound File to Your Game Using XACT](#).

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ChangeSoundVolume_Sample.zip](#).

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace ChangeSoundVolume
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        // Audio objects
        AudioEngine engine;
        SoundBank soundBank;
        WaveBank waveBank;
        AudioCategory musicCategory;

        // Music volume.
        float musicVolume = 1.0f;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        protected override void Initialize()
        {
            base.Initialize();

            // Initialize audio objects.
            engine = new AudioEngine("Content\\Audio\\ChangeSoundVolume.xgs");
            soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
        }
    }
}
```

```

        waveBank = new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb");

        // Get the category.
        musicCategory = engine.GetCategory("Music");

        // Play the sound.
        soundBank.PlayCue("music");
    }

    protected override void LoadContent()
    {
    }

    protected override void UnloadContent()
    {
    }

    protected void UpdateInput()
    {
        // Get the current gamepad state.
        GamePadState currentState = GamePad.GetState(PlayerIndex.One);

        if (currentState.DPad.Up == ButtonState.Pressed)
        {
            musicVolume = MathHelper.Clamp(musicVolume + 0.01f, 0.0f, 2.0f);
        }
        if (currentState.DPad.Down == ButtonState.Pressed)
        {
            musicVolume = MathHelper.Clamp(musicVolume - 0.01f, 0.0f, 2.0f);
        }

        // Set the category volume.
        musicCategory.SetVolume(musicVolume);
    }

    protected override void Update(GameTime gameTime)
    {
        // Allow the game to exit.
        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
            this.Exit();

        // Check input.
        UpdateInput();

        // Update the audio engine.
        engine.Update();

        base.Update(gameTime);
    }

    protected override void Draw(GameTime gameTime)
    {
        graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
        base.Draw(gameTime);
    }
}

```

Changing Sound Volume Levels

To change sound volume levels

1. Create an [AudioEngine](#), [WaveBank](#) and [SoundBank](#) at game start.
2. During game update, call the [Update](#) method of the [AudioEngine](#) to allow the audio engine to process audio data.
3. To retrieve a category of sounds whose volume you want to change, call [AudioEngine.GetCategory](#), and then pass in the name of the category you created in the XACT project.

4. Call [AudioCategory.SetVolume](#) on the retrieved category.

5. Specify the desired volume.

Typically, your preferred volume will be between 0.0f (silence) and 1.0f (full volume, as authored).

6. To make the volume louder than designed, pass in values greater than 1.0f.

For example, a volume of 2.0f adds +6 dB to the authored level.

How To: Stream a Sound Using XACT

Demonstrates how to create and use a streaming wave bank.

This topic builds on concepts in the [Audio Overview](#) topic, and [How To: Add a Sound File to Your Game Using XACT](#).

Note

This example assumes you already built an XACT sound bank and wave bank. To learn how to do this, see [How To: Add a Sound File to Your Game Using XACT](#).

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download StreamSound_Sample.zip](#).

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace StreamSound
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;

        // Audio objects
        AudioEngine engine;
        SoundBank soundBank;
        WaveBank waveBank;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";
        }

        protected override void Initialize()
        {
            base.Initialize();

            // Initialize audio objects.
            engine = new AudioEngine("Content\\Audio\\PlaySound.xgs");
            soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
            // Create streaming wave bank.
            waveBank =
                new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb", 0, 4);

            // must call update
            engine.Update();

            // Play the sound.
            soundBank.PlayCue("kaboom");
        }
    }
}
```

```

    }

    protected override void LoadContent()
    {
        // Create a new SpriteBatch, which can be used to draw textures.
        spriteBatch = new SpriteBatch(GraphicsDevice);

        // TODO: use this.Content to load your game content here
    }

    protected override void UnloadContent()
    {
    }

    protected override void Update(GameTime gameTime)
    {
        // Allows the game to exit
        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
            this.Exit();

        // Update the audio engine.
        engine.Update();

        base.Update(gameTime);
    }

    protected override void Draw(GameTime gameTime)
    {
        graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
        base.Draw(gameTime);
    }
}
}

```

Latency

In this context, *latency* is the amount of time that a device takes to retrieve waves from memory and subsequently play the sound. If latency is substantial, it can cause delays in audio playback, or possibly it can interrupt game sounds altogether.

Why Stream?

A wave bank may be designated as *in-memory*, which means all of the waves are loaded from the storage medium into game memory. Otherwise, a wave bank may be designated as *streaming*, which means that waves remain stored on disk. When a wave bank is designated as streaming, the sound information is delivered to the game through in-memory buffers. This is useful for sounds that are too large or too long to completely load into working memory. Streaming allows a game developer to effectively use large amounts of audio data without using too much memory.

When to Stream

Most often it is better to store wave banks in-memory. When waves are in-memory, the playback of the sound always occurs without latency. However, in cases when the sounds are too large to load into memory, a streaming wave bank may be a better choice. For example, this could be the case with background music that is much longer than other game sounds, such as short sound effects.

Creating a Streaming Wave Bank

Using XACT

The process for creating the streaming wave bank in XACT is the same as the process described in [How To: Add a Sound File to Your Game Using XACT](#), with the exception of one step. When creating the new wave bank, select the *streaming* option in the dialog box.

General

Name
Wave Bank

Notes
...

Type
In Memory Streaming 309.53 KB

Friendly Names Sync In-Game Data

Compression Preset
<none>

Paths

Header Path
...

Xbox 360 Build Path
Wave Bank.xwb

Windows Build Path
Wave Bank.xwb

In Code

You can create a streaming wave bank by using the [WaveBank \(AudioEngine, String, Int32, Int16\)](#) constructor that requires two streaming-related parameters: *offset* and *packet size*. An offset of zero will start the stream at the beginning of the wave file. Packet size determines how much memory a streaming wave bank uses for buffering. A smaller packet size will result in smaller amounts of data streaming through the buffer at any given time. A larger packet size indicates a larger amount of data moving through the buffer. Ideally, a game developer will specify a packet size that uses the least amount of memory, while at the same time allows for uninterrupted sounds during game play.

Note

Before you execute code to play any sounds, you must call the [Update](#) method of the [AudioEngine](#) that was used to create the streaming wave bank. Calling the **Update** method prepares the streaming wave bank for use. If you try to use the wave bank before you call the Update method, an **InvalidOperation** exception will be thrown.

See Also

Tasks

[How To: Add a Sound File to Your Game Using XACT](#)

[How To: Play a Sound Using XACT](#)

How To: Apply Basic 3D Positional Effects to a Cue

Demonstrates how to apply 3D positioning effects to cues.

The XNA Framework audio system contains support for 3D audio positioning effects. It uses the [AudioEmitter](#) and [AudioListener](#) classes, and the [Cue.Apply3D](#) method. The effects simulate 3D positioning for sound by adjusting speaker mix for cues that use the 3D values.

Note that speaker mix is the only effect that will be automatically applied using this method. Attenuation and Doppler-shift pitch modification effects must be applied via creation of runtime parameter controls (RPC) in the Microsoft Cross-Platform Audio Creation Tool (XACT). See [How To: Apply Attenuation and Doppler 3D Audio Effects](#) and [Apply3D](#) for more information.

The Complete Sample

The following example uses a circular rotation around a stationary [AudioListener](#) to emphasize the 3D effect.

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download Basic3DAudio_Sample.zip.](#)

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace Basic3DAudio
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        // Audio objects
        AudioEngine engine;
        SoundBank soundBank;
        WaveBank waveBank;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
        }

        // 3D audio objects
        AudioEmitter emitter = new AudioEmitter();
        AudioListener listener = new AudioListener();
        Cue cue;

        protected override void Initialize()
        {
            base.Initialize();

            // Initialize audio objects
            engine = new AudioEngine("Content\\Audio\\3DAudio.xgs");
            soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
            waveBank = new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb");

            // Get the cue and play it.
```

```

        // For 3D cues, you must call Apply3D before calling Play.
        cue = soundBank.GetCue("buzz");
        cue.Apply3D(listener, emitter);
        cue.Play();
    }

    protected override void LoadContent()
    {
    }

    protected override void UnloadContent()
    {
    }

    protected override void Update(GameTime gameTime)
    {
        // Allow the game to exit.
        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
            this.Exit();

        // Move the object around in a circle.
        Vector3 objectPos = new Vector3(
            (float)Math.Cos(gameTime.TotalGameTime.Seconds) / 2,
            0,
            (float)Math.Sin(gameTime.TotalGameTime.Seconds));

        // Apply 3D settings to the cue.
        emitter.Position = objectPos;
        cue.Apply3D(listener, emitter);

        // Update the audio engine
        engine.Update();

        base.Update(gameTime);
    }

    protected override void Draw(GameTime gameTime)
    {
        graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
        base.Draw(gameTime);
    }
}

```

Applying Basic 3D Positional Effects to a Cue

To retrieve and play a cue in 3D

1. Create an XACT project and add it to a new XNA Game Studio project as described in [How To: Add a Sound File to Your Game Using XACT](#).
The project should contain at least one cue.
2. In code, create an [AudioEngine](#), [WaveBank](#), and [SoundBank](#) at game start.
3. Create an [AudioEmitter](#), an [AudioListener](#), and a [Vector3](#) to store the 3D position of the sound entity.
4. In the [Game.Initialize](#) method, load the [AudioEngine](#), [WaveBank](#), and [SoundBank](#).
5. Call [SoundBank.GetCue](#) to retrieve the [Cue](#) you want to play in 3D.
6. Call [Cue.Apply3D](#) on the cue you retrieved in the previous step.
7. Call [Cue.Play](#) to begin playback of the cue.

Note that you must have called [Apply3D](#) on the cue first. Otherwise, the next call to [Apply3D](#) will throw an exception.

To process audio data

1. During game update, call the [Update](#) method of the [AudioEngine](#) to enable the audio engine to process audio data.
2. Set the [Vector3](#) to the position from which you want the sound to come.
3. Set the [AudioEmitter.Position](#) property to this vector.
4. As an option, set the [Vector3](#) to the position where you want the listener of the 3D sound to be, and then set the [AudioListener.Position](#) property to this vector.
5. Call [Cue.Apply3D](#) on the cue object you retrieved previously, passing in the [AudioEmitter](#), which is an [AudioListener](#).

Note

Calling the [Cue.Apply3D](#) method automatically sets the speaker mix for any sound played by this cue to a value calculated by the difference in [Position](#) values between *listener* and *emitter*. In preparation for the mix, the sound is converted to monoaural. Any stereo information in the sound is discarded.

How To: Apply Attenuation and Doppler 3D Audio Effects

Demonstrates how to apply attenuation and Doppler 3D positioning effects in code.

Introduction

This topic builds on [How To: Apply Basic 3D Positional Effects to a Cue](#), which discusses the steps required to add 3D audio to your game. This topic introduces concepts such as XACT projects, the audio engine, cues, and emitter and listener positions.

3D game audio typically implements at least three effects: speaker positioning, volume attenuation over distance, and Doppler pitch shifting. The basic 3D audio how-to discusses speaker positioning. This topic explains how to add attenuation and Doppler effects to 3D audio.

At design time, the sound designer defines the attenuation and Doppler effects and applies them to specific sounds. The designer uses XACT Runtime Parameter Controls (RPC) to specify the relationships between game information, exposed as XACT variables, and resulting sound effects. To determine final attenuation and Doppler effects at run time, the XACT engine combines the designer's effects with calculations based on position and velocity of the audio source and listener.

Sound designers can add attenuation without requiring any additional code changes by the developer (assuming that the game already sets the emitter and listener positions). However, Doppler effect requires that XACT know the velocity of the emitter and listener. Therefore, the game must set the [AudioEmitter.Velocity](#) and [AudioListener.Velocity](#) properties.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download Advanced3DAudio_Sample.zip](#).

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace Advanced3DAudio
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;

        // Audio objects
        AudioEngine engine;
        SoundBank soundBank;
        WaveBank waveBank;

        // 3D audio objects
        AudioEmitter emitter = new AudioEmitter();
        AudioListener listener = new AudioListener();
        Cue cue;

        float maxEmitterDistance = 150.0f;
        float maxVelocity = 30.0f;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
```



```

}

protected override void Initialize()
{
    base.Initialize();

    // Initialize audio objects.
    engine = new AudioEngine("Content\\Audio\\Advanced3DAudio.xgs");
    soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
    waveBank = new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb");

    // Set emitter and listener position.
    emitter.Position = Vector3.Backward;
    listener.Position = Vector3.Zero;

    // Get the cue and play it.
    // For 3D cues, you must call Apply3D before calling Play.
    cue = soundBank.GetCue("buzz");
    cue.Apply3D(listener, emitter);
    cue.Play();
}

protected override void LoadContent()
{
}

protected override void UnloadContent()
{
}

protected override void Update(GameTime gameTime)
{
    // Allow the game to exit.
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    // Move the sound left and right out to maximum distance.
    emitter.Position = new Vector3(
        (float)Math.Cos(gameTime.TotalGameTime.TotalSeconds / 5.0f) *
        maxEmitterDistance, 0.0f, 1.0f);

    // Add velocity with left or right triggers.
    emitter.Velocity = new Vector3(maxVelocity *
        GamePad.GetState(PlayerIndex.One).Triggers.Left, 0.0f, 0.0f);
    listener.Velocity = new Vector3(-maxVelocity *
        GamePad.GetState(PlayerIndex.One).Triggers.Right, 0.0f, 0.0f);

    // Apply 3D settings to the cue.
    cue.Apply3D(listener, emitter);

    // Update the audio engine.
    engine.Update();

    base.Update(gameTime);
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
    base.Draw(gameTime);
}
}
}

```

See Also

Concepts

[Audio Overview](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)
[Attenuation and Doppler Pitch Shifting Overview](#)
[Tutorial 3: Making Sounds with XNA Game Studio](#)
[XACT Runtime Parameter Controls \[DirectX\]](#)
[XACT Audio Authoring \[DirectX\]](#)

Reference

[Cue.Apply3D Method](#)
[AudioEmitter.Position Property](#)
[AudioListener.Position Property](#)
[AudioEmitter.Velocity Property](#)
[AudioListener.Velocity Property](#)
[AudioEmitter.DopplerScale Property](#)

Media

Describes how the XNA Framework [Microsoft.Xna.Framework.Media](#) namespace provides classes and methods for retrieving system media, including pictures and songs.

In This Section

[Media Overview](#)

Provides an introduction to media in XNA Game Studio.

[How To: Show Pictures](#)

Demonstrates how to use the [MediaLibrary](#) to access pictures in a picture album.

[How To: Play Video](#)

Demonstrates how to use the [VideoPlayer](#) to play back videos.

[How To: Play a Video in 3D space](#)

Demonstrates how to use the [VideoPlayer](#) to playback videos on the surface of a quad.

Media Overview

Describes how the [Microsoft.Xna.Framework.Media](#) namespace provides support for browsing or searching media items in XNA Game Studio. Media is a general term referring to all media on the local system.

Discovery and Enumeration

A game can query strongly-typed media collections for songs, artists, albums, playlist, genres, pictures, and picture albums using the [Microsoft.Xna.Framework.Media](#) namespace. See [How To: Show Pictures](#) for an example. Items in the media library can also be searched based on metadata such as artist name, album name, or musical genre.

Songs as Background Music

Access to the media library, combined with the ability to use playlists, allows games to create interesting background scores that can change with game play. Songs can be played directly from the media library, or imported using the content pipeline.

Pictures as Textures

Games can obtain and use textures from pictures within a media library.

Video Playback

Games can now add videos to their projects using the content pipeline, and display those videos on screen. These videos can be displayed as cutscenes or loading graphics, or as part of an object in 3D space.

To add a video to a project, use the steps outlined in "How To Load Content," and select a WMV file to import. The video content processor that comes with XNA Game Studio has a few limitations:

1. Videos with Digital Rights Management enabled cannot be used.
2. The video must be encoded to the WMV-9 "Main" profile using the VC-1 standard.
3. The video must be encoded with a constant bit rate (CBR).
4. There must be an audio track in the video.
5. The audio accompanying the video must be Windows Media Audio (WMA) encoded, with a single-pass CBR format.
6. XNA Game Studio supports the following maximum bit rates:

Profile	Level	Maximum Bit Rate	Representative Resolutions by Frame Rate
Main	Low	2 Mbps	320 × 240 @ 24 Hz (QVGA)
	Medium	10 Mbps	720 × 480 @ 30 Hz (480p) 720 × 576 @ 25 Hz (576p)
	High	20 Mbps	1280 × 720 @ 30 Hz (720p)

A video must have an audio track or it will fail to load. If it fails to load, this message appears: "Video file is invalid. Please make sure that the video is not DRM protected and is a valid single-pass CBR encoded video file."

Once a video is loaded, games can create a new [VideoPlayer](#) object to play, pause, and resume the video. Once the video has started playing, the game can call the [GetTexture](#) method to get a [Texture2D](#) containing the current frame of video. That texture can be used by [SpriteBatch.Draw](#) or rendered by an Effect as part of an object in 3D space.

How To: Show Pictures

Demonstrates how to use the [MediaLibrary](#) to access pictures in a picture album.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download ShowPictures.zip](#).

Accessing Pictures

This example retrieves all pictures in a user's picture library.

Declare variables.

C#

```
ICollection<MediaSource> mediaSources;  
MediaLibrary mediaLib;  
PictureCollection picCollection;
```

Access the media library.

C#

```
mediaSources = MediaSource.GetAvailableMediaSources();  
foreach (MediaSource ms in mediaSources)  
{  
    mediaLib = new MediaLibrary(ms);  
}  
picCollection = mediaLib.Pictures;
```

See Also

Reference

[MediaPlayer Class](#)

[MediaLibrary Class](#)

How To: Play Video

Demonstrates how to use the [VideoPlayer](#) to play back videos.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download VideoPlayback_Sample.zip.](#)

Playing Video

To play a video

Add a video to the game project using the steps outlined in [How To: Load Content](#).

1. In [LoadContent](#), load the video using the [ContentManager](#).
2. Create a new instance of the [VideoPlayer](#).

C#

```
Video video;
VideoPlayer player;
Texture2D videoTexture;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    video = Content.Load<Video>("video");
    player = new VideoPlayer();
}
```

3. In the [Update](#) method, play the video. Set [IsLooped](#) to **true** if you want the video to repeat.

C#

```
protected override void Update(GameTime gameTime)
{
    ...
    if (player.State == MediaState.Stopped)
    {
        player.IsLooped = true;
        player.Play(video);
    }
    ...
    base.Update(gameTime);
}
```

4. In [Draw](#), if the [VideoPlayer](#) is not stopped, call [GetTexture](#) to get a copy of the latest frame of video.
5. Create a [Rectangle](#) defining where on-screen the video will appear.
6. If the texture is not **null**, draw it onscreen using [SpriteBatch.Draw](#).

C#

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    // Only call GetTexture if a video is playing or paused
    if (player.State != MediaState.Stopped)
        videoTexture = player.GetTexture();
}
```

```
// Drawing to the rectangle will stretch the
// video to fill the screen
Rectangle screen = new Rectangle(GraphicsDevice.Viewport.X,
    GraphicsDevice.Viewport.Y,
    GraphicsDevice.Viewport.Width,
    GraphicsDevice.Viewport.Height);

// Draw the video, if we have a texture to draw.
if (videoTexture != null)
{
    spriteBatch.Begin();
    spriteBatch.Draw(videoTexture, screen, Color.White);
    spriteBatch.End();
}

base.Draw(gameTime);
}
```

See Also

Concepts

[Media Overview](#)

Tasks

[How To: Draw a Sprite](#)

Reference

[Video Class](#)

[VideoPlayer Class](#)

How To: Play a Video in 3D space

Demonstrates how to use the [VideoPlayer](#) to playback videos on the surface of a quad.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download VideoQuad_Sample.zip.](#)

Playing Video

To play a video

1. Create a quad (or other 3D surface) on which to display the video.

This sample uses the Quad class from [How To: Draw a Textured Quad](#).

2. Create a View and Projection matrix for viewing the quad.

C#

```
Quad quad;
Matrix View, Projection;
protected override void Initialize()
{
    // Create a scene, including a quad and a view matrix looking at it
    quad = new Quad(Vector3.Zero, Vector3.Backward, Vector3.Up, 1, 1);
    View = Matrix.CreateLookAt(new Vector3(0, 0, 2),
        Vector3.Zero, Vector3.Up);
    Projection = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.PiOver4, 4.0f / 3.0f, 1, 500);

    base.Initialize();
}
```

3. Add a video to the game project using the steps outlined in [How To: Load Content](#).
4. In [LoadContent](#), use the [ContentManager](#) to load the video.
5. Create a new instance of the [VideoPlayer](#).

Set [IsLooped](#) to **true** if you want the video to repeat.

6. Set up your [Effect](#) for drawing.

In this case, we use [BasicEffect](#), enabling default lighting and a texture.

C#

```
BasicEffect quadEffect;
Video video;
VideoPlayer player;
Matrix World;
VertexDeclaration quadVertexDecl;
protected override void LoadContent()
{
    // Load a video, and initialize a player
    video = Content.Load<Video>("video");
    player = new VideoPlayer();
    player.IsLooped = true;

    // Setup our BasicEffect for drawing the quad
    World = Matrix.CreateScale((float)video.Width, (float)video.Height, 1, 1);
}
```



```

quadEffect = new BasicEffect(graphics.GraphicsDevice, null);
quadEffect.EnableDefaultLighting();
quadEffect.World = World;
quadEffect.View = View;
quadEffect.Projection = Projection;
quadEffect.TextureEnabled = true;

// Create a vertex declaration so we can call
// DrawUserIndexedPrimitives later
quadVertexDecl = new VertexDeclaration(GraphicsDevice,
    VertexPositionNormalTexture.VertexElements);
}

```

7. In the `Update` method, play the video.

C#

```

protected override void Update(GameTime gameTime)
{
    ...
    // Play the video if it isn't already.
    if (player.State != MediaState.Playing)
        player.Play(video);

    base.Update(gameTime);
}

```

8. In `Draw`, if the `VideoPlayer` is not stopped, call `GetTexture` to get a copy of the latest frame of video, and link it to your effect.

9. Draw the 3D object using your effect.

C#

```

protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    // If the video is playing, get the current frame
    // as a texture. (Calling GetTexture on a stopped
    // player results in an exception)
    if (player.State == MediaState.Playing)
        quadEffect.Texture = player.GetTexture();

    // Draw the quad
    GraphicsDevice.VertexDeclaration = quadVertexDecl;
    quadEffect.Begin();
    foreach (EffectPass pass in quadEffect.CurrentTechnique.Passes)
    {
        pass.Begin();

        GraphicsDevice.DrawUserIndexedPrimitives
            <VertexPositionNormalTexture>(
                PrimitiveType.TriangleList,
                quad.Vertices, 0, 4,
                quad.Indexes, 0, 2);

        pass.End();
    }
    quadEffect.End();
}

```

```
    base.Draw(gameTime);  
}
```

See Also

Concepts

[Media Overview](#)

Tasks

[How To: Play a Video in 3D space](#)

[How To: Draw a Textured Quad](#)

Reference

[Video Class](#)

[VideoPlayer Class](#)

Storage

Provides classes that allow reading and writing of files.

In This Section

[Storage Overview](#)

The XNA Framework provides a set of classes in the [Microsoft.Xna.Framework.Storage](#) namespace that are used to read and write files from user storage and title storage, including player saves or title data.

[How To: Create a File](#)

Demonstrates how to use the [StorageContainer](#) class to create a save game file in the title storage area on a device specified by the gamer.

[How To: Open a File](#)

Demonstrates how to use the [StorageContainer](#) class to open a save game file in the title storage area on a device specified by the gamer.

[How To: Copy a File](#)

Demonstrates how to use the [StorageContainer](#) class to copy a save game file in the title storage area on a device specified by the gamer.

[How To: Rename a File](#)

Demonstrates how to use the [StorageContainer](#) class to rename a save game file in the title storage area on a device specified by the gamer.

[How To: Enumerate Files](#)

Demonstrates how to use the [StorageContainer](#) class to get a list of save game files in the title storage area on a device specified by the gamer.

[How To: Delete a File](#)

Demonstrates how to use the [StorageContainer](#) class to delete a save game file in the title storage area on a device specified by the gamer.

[How To: Load a Game Data File](#)

Demonstrates how to use the [File](#) class to read a file from the game image directory.

[How To: Serialize Data](#)

Demonstrates how to use the [XmlSerializer](#) class to write data in a custom class to a saved game, and how to load the data from the file.

[How To: Get a StorageDevice Asynchronously](#)

Demonstrates how to call [BeginShowStorageDeviceSelector](#) and [EndShowStorageDeviceSelector](#) to get a [StorageDevice](#) object asynchronously.

Storage Overview

The XNA Framework provides a set of classes in the [Microsoft.Xna.Framework.Storage](#) namespace that are used to read and write files from user storage and title storage, including player saves or title data.

- [Storage Devices](#)
- [Storage Locations](#)

Storage Devices

When you are programming with the XNA Framework, you read files from two general locations:

- Title storage, where the game executable and support files (shaders, meshes, textures, and so on) are located.
- User storage, the space a player provides for game data (such as saves) that is either locked to a particular profile or available to all players.

To access title storage, the XNA Framework provides the [StorageContainer](#) class, whose [TitleLocation](#) property contains a path to the location of the title itself. Title support files can be loaded based on this location.

To access user storage, the XNA Framework provides the [Guide](#) class, which contains the [BeginShowStorageDeviceSelector](#) method. This method brings up a dialog box on Xbox 360 where the user selects a storage device (memory unit or hard drive). This method works asynchronously, notifying the caller with an [IAsyncResult](#) or a callback method when the user has chosen a device. For more information, see [How To: Get a StorageDevice Asynchronously](#).

The [StorageContainer](#) instance provided by [OpenContainer](#) provides access to the selected device. Use **System.IO** objects, such as the [File](#) object, to open, copy, rename, or delete files, using [StorageContainer.Path](#) to determine the location of the files on the device.

Changes to the files within a [StorageContainer](#) on Xbox 360 do not take effect until the [StorageContainer](#) is disposed. For this reason, only one [StorageContainer](#) can be open on the same device for the same profile. Before opening a [StorageContainer](#) on a device, make sure that all previous [StorageContainer](#) objects opened for that profile have been disposed. To dispose a [StorageContainer](#), call [Dispose](#).

Storage Locations

Currently, the title location on a PC is the folder where the executable resides when it is run. Use the [TitleLocation](#) property to access the path.

User storage is in the My Documents folder of the user who is currently logged in, in the **SavedGames** folder. A subfolder is created for each game according to the **titleName** passed to the [OpenContainer](#) method. When no [PlayerIndex](#) is specified, content is saved in the **AllPlayers** folder. When a [PlayerIndex](#) is specified, the content is saved in the **Player1**, **Player2**, **Player3**, or **Player4** folder, depending on which [PlayerIndex](#) was passed to [BeginShowStorageDeviceSelector](#).

See Also

Tasks

- [How To: Create a File](#)
- [How To: Open a File](#)
- [How To: Copy a File](#)
- [How To: Rename a File](#)
- [How To: Enumerate Files](#)
- [How To: Delete a File](#)
- [How To: Load a Game Data File](#)
- [How To: Serialize Data](#)
- [How To: Get a StorageDevice Asynchronously](#)

Reference

- [StorageDevice](#)
- [BeginShowStorageDeviceSelector](#)
- [StorageContainer](#)
- [TitleLocation](#)
- [Path](#)

How To: Create a File

Demonstrates how to use the [StorageContainer](#) class to create a save game file in the title storage area on a device specified by the gamer. This example assumes you obtained a [StorageDevice](#). To obtain a [StorageDevice](#), see [How To: Get a StorageDevice Asynchronously](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip](#).

Creating a File

To create a save game file in title storage

1. Call the [Guide.BeginShowStorageDeviceSelector](#) method to get a device index indicating which device the user prefers.
2. Open a [StorageContainer](#) on the device, and then pass the name of your title.
3. Call [Path.Combine](#) to merge the container path with the name of the file to be created.
4. Call [File.Exists](#) to confirm that the file is not already present.
5. Call [File.Create](#) with the file name of the file to create.

[File.Create](#) returns a [FileStream](#) object that you can use to add data to the file.

6. Call [Close](#) to close the file after you have populated it with data.
7. Call [StorageContainer](#) to commit the changes to the device.

C#

```
private static void DoCreate(StorageDevice device)
{
    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");

    // Add the container path to our file name.
    string filename = Path.Combine(container.Path, "demobinary.sav");

    // Create a new file.
    if (!File.Exists(filename))
    {
        FileStream file = File.Create(filename);
        file.Close();
    }
    // Dispose the container, to commit the data.
    container.Dispose();
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice](#)

[BeginShowStorageDeviceSelector](#)

[StorageContainer](#)

[Path](#)

How To: Open a File

Demonstrates how to use the [StorageContainer](#) class to open a save game file in the title storage area on a device specified by the gamer. This example assumes you obtained a [StorageDevice](#). To obtain a [StorageDevice](#), see [How To: Get a StorageDevice Asynchronously](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip](#).

Opening a File

To open a save game file in title storage

1. Call the [Guide.BeginShowStorageDeviceSelector](#) method to get a device index indicating which device the user prefers.
2. Open a [StorageContainer](#) on the device, and then pass the name of your title.
3. Call [Path.Combine](#) to merge the container path with the name of the file to be opened.
4. Call [File.Open](#) with the file name of the file to open.

[File.Open](#) returns a [FileStream](#) object that you can use to add data to the file.

5. Call [Close](#) to close the file after you have read it.
6. Dispose the [StorageContainer](#) after closing the file.

C#

```
private static void DoOpen(StorageDevice device)
{
    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");

    // Add the container path to our file name.
    string filename = Path.Combine(container.Path, "demobinary.sav");

    FileStream file = File.Open(filename, FileMode.Open);
    file.Close();

    // Dispose the container.
    container.Dispose();
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice](#)

[BeginShowStorageDeviceSelector](#)

[StorageContainer](#)

[Path](#)

How To: Copy a File

Demonstrates how to use the [StorageContainer](#) class to copy a save game file in the title storage area on a device specified by the gamer. This example assumes you already obtained a [StorageDevice](#). To obtain a [StorageDevice](#), see [How To: Get a StorageDevice Asynchronously](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip](#).

Copying a File

To copy a save game file in title storage

1. Call the [Guide.BeginShowStorageDeviceSelector](#) method to get a device index indicating which device the user prefers.
2. Open a [StorageContainer](#) on the device, and then pass the name of your title.
3. Call [Path.Combine](#) to merge the container path with the name of the source and destination files.
4. Call [File.Copy](#) with the file name of the file to copy, and the name of the new copy to be created.

You may also specify whether to overwrite existing files.

5. Dispose the [StorageContainer](#) to commit the changes to the device.

C#

```
private static void DoCopy(StorageDevice device)
{
    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");

    // Add the container path to our file name.
    string filename = Path.Combine(container.Path, "demobinary.sav");
    string copyfilename = Path.Combine(container.Path, "copybinary.sav");

    File.Copy(filename, copyfilename, true);

    // Dispose the container, to commit the change.
    container.Dispose();
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice](#)

[BeginShowStorageDeviceSelector](#)

[StorageContainer](#)

[Path](#)

How To: Rename a File

Demonstrates how to use the [StorageContainer](#) class to rename a save game file in the title storage area on a device specified by the gamer. This example assumes you obtained a [StorageDevice](#). To obtain a [StorageDevice](#), see [How To: Get a StorageDevice Asynchronously](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip](#).

Renaming a File

To rename a save game file in title storage

1. Call the [Guide.BeginShowStorageDeviceSelector](#) method to get a device index indicating which device the user prefers.
2. Open a [StorageContainer](#) on the device, and then pass the name of your title.
3. Call [Path.Combine](#) to merge the container path with the name of the source and destination files.
4. Call [File.Exists](#) to confirm that the destination file does not exist.
5. Call [File.Move](#) with the name of the file to rename, and the new name of the file.
6. Dispose the [StorageContainer](#) to commit the changes to the device.

C#

```
private static void DoRename(StorageDevice device)
{
    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");

    // Add the container path to our file name.
    string oldfilename = Path.Combine(container.Path, "demobinary.sav");
    string newfilename = Path.Combine(container.Path, "renamebinary.sav");

    if (!File.Exists(newfilename))
        File.Move(oldfilename, newfilename);

    // Dispose the container, to commit the change
    container.Dispose();
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice](#)

[BeginShowStorageDeviceSelector](#)

[StorageContainer](#)

[Path](#)

How To: Enumerate Files

Demonstrates how to use the [StorageContainer](#) class to get a list of save game files in the title storage area on a device specified by the gamer. This example assumes you obtained a [StorageDevice](#). To obtain a [StorageDevice](#), see [How To: Get a StorageDevice Asynchronously](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip](#).

Enumerating Files

To enumerate save game files in title storage

1. Call the [Guide.BeginShowStorageDeviceSelector](#) method to get a device index indicating which device the user prefers.
2. Open a [StorageContainer](#) on the device, and then pass the name of your title.
3. Call [Directory.GetFiles](#).

[Directory.GetFiles](#) returns a collection of strings, where each string is the name of a file in the container.

4. Dispose the [StorageContainer](#).

C#

```
private static void DoEnumerate(StorageDevice device)
{
    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");

    ICollection<string> FileList = Directory.GetFiles(container.Path);
    foreach (string filename in FileList)
    {
        Console.WriteLine(filename);
    }

    // Dispose the container.
    container.Dispose();
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice](#)

[BeginShowStorageDeviceSelector](#)

[StorageContainer](#)

[Path](#)

How To: Delete a File

Demonstrates how to use the [StorageContainer](#) class to delete a save game file in the title storage area on a device specified by the gamer. This example assumes you obtained a [StorageDevice](#). To obtain a [StorageDevice](#), see [How To: Get a StorageDevice Asynchronously](#).

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip](#).

Deleting a File

To delete a saved game file in title storage

1. Call the [Guide.BeginShowStorageDeviceSelector](#) method to get a device index indicating which device the user prefers.
2. Open a [StorageContainer](#) on the device, and then pass the name of your title.
3. Call [Path.Combine](#) to merge the container path with the name of the file to be deleted.
4. Call [File.Exists](#) to confirm that the file is present.
5. Call [File.Delete](#) with the file name of the file to delete.
6. Call the [StorageContainer](#) method to commit the changes to the device.

C#

```
private static void DoDelete(StorageDevice device)
{
    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");

    // Add the container path to our file name.
    string filename = Path.Combine(container.Path, "demobinary.sav");

    // Delete the new file.
    if (File.Exists(filename))
        File.Delete(filename);

    // Dispose the container, to commit the change.
    container.Dispose();
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice](#)

[BeginShowStorageDeviceSelector](#)

[StorageContainer](#)

[Path](#)

How To: Load a Game Data File

Demonstrates how to use the [File](#) class to read a file from the game image directory.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip.](#)

Loading a Game Data File

To load a game data file

1. Construct the full path to the file by combining [StorageContainer.TitleLocation](#) with the file name.
2. Call [File.Open](#) with the full file path to get a [FileStream](#) containing the file data.

C#

```
/// <summary>
/// This method opens a file using System.IO classes and the
/// TitleLocation property. It presumes that a file named
/// ship.dds has been deployed alongside the game.
/// </summary>
private static void DoOpenFile()
{
    FileStream file = OpenTitleFile(
        "ship.dds", FileMode.Open, FileAccess.Read);
    Console.WriteLine("File Size: " + file.Length);
    file.Close();
}
private static FileStream OpenTitleFile(
    string filename, FileMode mode, FileAccess access)
{
    string fullpath = Path.Combine(StorageContainer.TitleLocation, filename);
    return File.Open(fullpath, mode, access);
}
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageContainer](#)

[TitleLocation](#)

How To: Serialize Data

Demonstrates how to use the [XmlSerializer](#) class to write data in a custom class to a saved game, and how to load the data from the file.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip.](#)

Serializing Data

To define save game data

1. Define a new class or structure.

The size of this class will determine the size of your saved game files, so try to keep the class small. Do not reference other objects from this class unless you want to serialize them as well.

2. Add the `Serializable` attribute to the class.

C#

```
[Serializable]
public struct SaveGameData
{
    public string PlayerName;
    public Vector2 AvatarPosition;
    public int Level;
    public int Score;
}
```

To serialize data to a save game file

1. Create a [StorageContainer](#) to access the specified device.
2. Use [Path.Combine](#) to create a new string specifying the full path to the save game file.
3. Open a [FileStream](#) object on the file using the [File.Open](#) method from the `System.IO` namespace.

Specifying **FileMode.Create** tells [Open](#) to create the save game if it doesn't exist. If it does, the file will be truncated to zero bytes.

4. Create an [XmlSerializer](#) object, passing the type of the structure that defines your save game data.
5. Call [Serialize](#), and then pass the [FileStream](#) and the data to serialize.

The [XmlSerializer](#) will turn the data in the structure into XML and use the [FileStream](#) to write the data into the file.

6. Close the [FileStream](#).
7. Dispose the [StorageContainer](#) to commit the changes to the device.

C#

```
private static void DoSaveGame(StorageDevice device)
{
    // Create the data to save.
    SaveGameData data = new SaveGameData();
    data.PlayerName = "Hiro";
    data.AvatarPosition = new Vector2(360, 360);
    data.Level = 11;
    data.Score = 4200;

    // Open a storage container.
    StorageContainer container =
        device.OpenContainer("StorageDemo");
}
```

```
// Get the path of the save game.
string filename = Path.Combine(container.Path, "savegame.sav");

// Open the file, creating it if necessary.
FileStream stream = File.Open(filename, FileMode.Create);

// Convert the object to XML data and put it in the stream.
XmlSerializer serializer = new XmlSerializer(typeof(SaveGameData));
serializer.Serialize(stream, data);

// Close the file.
stream.Close();

// Dispose the container, to commit changes.
container.Dispose();
}
```

To read serialized data from a save game file

1. Create a [StorageContainer](#) to access the specified device.
2. Use [Path.Combine](#) to create a new string specifying the full path to the save game file.
3. Call [File.Exists](#) to determine if the save game exists.
4. Open a [FileStream](#) object on the file using the [File.Open](#) method from the System.IO namespace.
5. Create an [XmlSerializer](#) object, and then pass the type of the structure that defines your save game data.
6. Call [Deserialize](#), and then pass the [FileStream](#) object.

[Deserialize](#) returns a copy of the save game structure populated with the data from the save game file. (You will have to cast the return value from Object to your type.)

7. Close the [FileStream](#).
8. Dispose the [StorageContainer](#).

See Also

Concepts

[Storage Overview](#)

Reference

[StorageContainer](#)

[Path](#)

How To: Get a StorageDevice Asynchronously

Demonstrates how to call [BeginShowStorageDeviceSelector](#) and [EndShowStorageDeviceSelector](#) to get a [StorageDevice](#) object asynchronously.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download StorageDemo_Sample.zip.](#)

Getting a StorageDevice Asynchronously

To get a StorageDevice asynchronously using IsCompleted

1. In your [Game](#) constructor, add a new [GamerServicesComponent](#) to your [Components](#) collection.
2. Create a variable to track when a request for a [StorageDevice](#) is pending. In this case, declare **GameSaveRequested**.

When the user needs to access the [StorageDevice](#) for the first time (for example, to save a game), call [Guide.BeginShowStorageDeviceSelector](#), specifying which player has requested the save.

3. Check [Guide.IsVisible](#) first to make sure the Guide is not already being displayed.

Calling [Guide.BeginShowStorageDeviceSelector](#) when the Guide is visible results in a [GuideAlreadyVisibleException](#). [Guide.BeginShowStorageDeviceSelector](#) returns an [IAsyncResult](#) interface that is used to determine when the asynchronous request is finished.

4. Set your tracking variable to indicate a request is pending.
5. When a request is pending, check [IAsyncResult.IsCompleted](#) periodically to determine when the user selected the storage device.
6. When [IsCompleted](#) is **true**, call [Guide.EndShowStorageDeviceSelector](#), passing the [IAsyncResult](#) provided by [Guide.BeginShowStorageDeviceSelector](#).

The return value is the selected storage device.

7. Use [IsConnected](#) to ensure that a valid device was chosen (if the user declines to select a device, or the device is removed, [IsConnected](#) will be **false**).
8. If the [StorageDevice](#) is connected, use it to load or save data.
9. Reset your tracking variable.

C#

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    this.Components.Add(new GamerServicesComponent(this));
}
IAsyncResult result;
Object stateobj;
bool GameSaveRequested = false;
GamePadState currentState;

protected override void Update(GameTime gameTime)
{
    GamePadState previousState = currentState;
    currentState = GamePad.GetState(PlayerIndex.One);
    // Allows the default game to exit on Xbox 360 and Windows
    if (currentState.Buttons.Back == ButtonState.Pressed)
```

```

        this.Exit();

    if ((currentState.Buttons.A == ButtonState.Pressed) &&
        (previousState.Buttons.A == ButtonState.Released))
    {
        // Set the request flag
        if ((!Guide.IsVisible) && (GameSaveRequested == false))
        {
            GameSaveRequested = true;
            result = Guide.BeginShowStorageDeviceSelector(PlayerIndex.One,
                null, null);
        }
    }
    // If a save is pending, save as soon as the
    // storage device is chosen
    if ((GameSaveRequested) && (result.IsCompleted))
    {
        StorageDevice device = Guide.EndShowStorageDeviceSelector(result);
        if (device != null && device.IsConnected)
        {
            DoSaveGame(device);
        }
        // Reset the request flag
        GameSaveRequested = false;
    }
    base.Update(gameTime);
}

```

To get a StorageDevice asynchronously using an AsyncCallback

1. In your `Game` constructor, add a new `GamerServicesComponent` to your `Components` collection.
2. Create an `AsyncCallback` object that represents the method that will be called when the player chooses a device. That function must take an `IAsyncResult` as a parameter, and return **void**.
3. When the user needs to access the `StorageDevice` for the first time (for example, to save a game), call `Guide.BeginShowStorageDeviceSelector`, specifying which player has requested the save, and your `AsyncCallback` object. You may optionally pass a tracking object to identify the request (or **null**).
4. Check `Guide.IsVisible` first to make sure the guide is not already being displayed.
5. Call `Guide.BeginShowStorageDeviceSelector` when the guide is visible results in a `GuideAlreadyVisibleException`.
6. In your callback method, call `Guide.EndShowStorageDeviceSelector`, passing the same `IAsyncResult` that was passed into the callback method. The return value is the selected storage device.
7. Use `IsConnected` to ensure that a valid device was chosen (if the user declines to select a device, or the device is removed, `IsConnected` will be **false**).
8. If the `StorageDevice` is connected, use it to load or save data.

C#

```

public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    this.Components.Add(new GamerServicesComponent(this));
}

```

```

IAsyncResult result;
Object stateobj;
bool GameSaveRequested = false;
GamePadState currentState;

protected override void Update(GameTime gameTime)
{
    GamePadState previousState = currentState;
    currentState = GamePad.GetState(PlayerIndex.One);
    // Allows the default game to exit on Xbox 360 and Windows
    if (currentState.Buttons.Back == ButtonState.Pressed)
        this.Exit();

    if ((currentState.Buttons.B == ButtonState.Pressed) &&
        (previousState.Buttons.B == ButtonState.Released))
    {
        if (!Guide.IsVisible)
        {
            // Reset the device
            device = null;
            stateobj = (Object)"GetDevice for Player One";
            Guide.BeginShowStorageDeviceSelector(
                PlayerIndex.One, this.GetDevice, stateobj);
        }
    }
    base.Update(gameTime);
}

StorageDevice device;
void GetDevice(IAsyncResult result)
{
    device = Guide.EndShowStorageDeviceSelector(result);
    if (device != null && device.IsConnected)
    {
        DoSaveGame(device);
    }
}

```

See Also

Concepts

[Storage Overview](#)

[Xbox 360 Programming Considerations](#)

Reference

[StorageDevice](#)

[Guide.BeginShowStorageDeviceSelector](#)

[Guide.EndShowStorageDeviceSelector](#)

[AsyncCallback](#)

[IAsyncResult](#)

Gamer Services

Contains introductory articles describing how to use gamer services: working with player profiles and preferences, the Xbox Guide user interface, Guide-based messaging, and other features provided by Xbox LIVE.

Overviews

[Gamer Services Overview](#)

Describes the functionality provided by the XNA Framework [GamerServices namespace](#), which includes querying for LIVE accounts, working with gamer Avatars, retrieving player preferences for local accounts, and showing various Xbox LIVE Guide user interface screens programmatically.

[Programming with Avatars](#)

Discusses the support for avatars in XNA Game Studio applications targeting the Xbox 360 console.

Gamer Services How-Tos

[How To: Use Gamertags and Gamer Pictures](#)

Demonstrates how to retrieve gamertags and pictures using a technique that can be applied to retrieve other gamer profile information.

[How To: Add Presence Information](#)

Describes how to configure and set presence information for a player.

[How To: Add Support for Game Invitations](#)

Describes how to support game invitations for Xbox LIVE Indie Games.

[How To: Add LIVE Party Support](#)

Describes how to add LIVE Party support to Xbox LIVE Indie Games.

[How To: Initialize and Update the Gamer Services Dispatcher](#)

Describes how an application would use the [GamerServicesDispatcher](#) instead of the [GamerServicesComponent](#) for cases in which a program may not be using the XNA Framework application model or component infrastructure.

[How To: Detect or Simulate Trial Mode and Present a Marketplace Offer](#)

Demonstrates how to detect and simulate trial mode for the purpose of testing the display of a Marketplace offer that will allow a player to purchase your game.

[How To: Render and Animate an Avatar Using AvatarRenderer.](#)

Demonstrates how to render and animate a gamer's avatar using the [AvatarRenderer](#) class and a standard animation.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Gamer Services Overview

The [GamerServices namespace](#) provides functionality available from Xbox LIVE to XNA Framework developers. This includes the capability to retrieve player preferences for local accounts, use gamer Avatars, and to display various LIVE Guide user interface screens programmatically.

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

- [Initialization of Gamer Services](#)
- [Player Sign-In](#)
- [Gamer Avatar](#)
- [Player Profile Information](#)
- [Guide User Interface](#)

Initialization of Gamer Services

Users must explicitly initialize the gamer services subsystem before they can use any gamer services or networking API calls. Most games will use the [GamerServicesComponent](#), a game component that takes care of the process of initializing and pumping the gamer services system. This component will pass through the graphics device and window handle automatically. It calls [Update](#) to pump the gamer services pump at regular intervals. To take advantage of this component, most games simply need to add one line to the [Game](#) constructor:

C#

```
Components.Add(new GamerServicesComponent(this));
```

Initializing the gamer services system can be a slow operation, and it has user-visible consequences. For example, the game may display the "Player Signed In" notification when the gamer services system is initialized. This happens when a local profile is set to sign in automatically, and, as a result, trigger a system update. Because of this, if a game is going to use these services, it is desirable to initialize everything in the game startup code in the constructor.

Programs that do not use the XNA Framework application model or component infrastructure can use [GamerServicesDispatcher](#). In this case, you must also call the [GamerServicesDispatcher.Initialize](#) and [GamerServicesDispatcher.Update](#) methods directly. Under these conditions, you should initialize the [GamerServicesDispatcher](#) once at startup, and call [GamerServicesDispatcher.Update](#) at every frame.

The [GamerServicesDispatcher.Update](#) method enables the XNA Framework to raise events for notifications such as sign-in status changes and game invites when it is safe to raise the events.

Player Sign-In

For standard game clients with a user interface, the Guide user interface automatically displays the appropriate user interface when a player presses either the HOME key on the keyboard or the **Guide** button on the Xbox 360 controller, allowing players to sign in. Before a player signs in, you can also choose to manually display the sign-in user interface using the [Guide.ShowSignIn](#) function.

When a [SignedInGamer](#) signs in, the [SignedInGamer.SignedIn](#) event occurs. You can get the list of signed-in players at any time using the static property [Gamer.SignedInGamers](#).

Gamer Avatar

Avatars are three-dimensional animated characters that represent the gamer in a versatile, personalized, engaging manner. Gamer services implements support for displaying and animating a gamer's avatar using standard effects and animations. In addition, access to the metadata of an avatar (such as hierarchy information and textures) allows developers to implement custom rendering and animating of a gamer's avatar.

Note

Avatar support is only available for Xbox 360 applications.

Avatars are used in a variety of ways on the Xbox 360 console: as a gamerpic and integrated into the friends list display, the Xbox Guide, and the dashboard. Also, game titles can use avatars in-game to represent various game players and to provide unique spectators, cut scene actors, crowd members, or game characters.

For more information on implementing avatar support in your application, see [Avatars](#).

Player Profile Information

The player profile information that is accessible using the [Gamer.GetProfile](#) method returns information such as the gamer picture, motto, and gamerzone. Specifically, the gamer profile provides the following information about a gamer:

- [Gamerscore](#)
- [Gamerzone](#)
- [Gamer picture](#)
- [Region](#)
- [Motto](#)
- [Reputation](#)
- [Number of achievements](#)
- [Number of titles played](#)

You may use [SignedInGamer.GameDefaults](#) to retrieve a local gamer's preferred settings, such as:

- [Controller sensitivity](#)
- [Game difficulty](#)
- [Primary](#) and [secondary](#) colors
- Action movement preferences such as [auto-aim](#), [auto-center](#), and [y-axis inversion](#)
- Preferred [racing camera angle](#), [accelerator](#), [brake](#), and [transmission](#)

Using [SignedInGamer.Privileges](#), you can see if the player is allowed to:

- [Communicate using voice, text, or messaging.](#)
- [Join online multiplayer sessions.](#)
- [View profiles.](#)

You can also view whether a gamer is [LIVE-enabled](#) or a [guest](#).

Privileges will be enforced by the framework. A player can never create an online session if his or her multiplayer privilege is not enabled. We provide programmatic access to these settings so that games may check to see if anything is forbidden. In such a case, the relevant menu options are grayed out—this keeps the user from selecting a forbidden option and getting an exception or error when trying to call the forbidden method.

Guide User Interface

The XNA Framework exposes a property, [Guide.IsVisible](#), which checks to see if the Guide user interface is active. The Guide runs independently alongside games. The player can view the Guide at any time by pressing the HOME key on the keyboard or the **Guide** button on the controller.

Many of the Guide calls are blocking. The XNA Framework uses the standard .NET async pattern to expose asynchronous versions of all the Guide methods. Blocking versions of the Guide methods automatically pump the graphics device while the UI is active. They call [GraphicsDevice.Present](#) at regular intervals with an empty (black) backbuffer. Games should have the asynchronous versions of the methods if it is desirable to continue rendering behind the Guide. The existing storage device UI has been changed to match the new design.

Programming with Avatars

Discusses the support for avatars in XNA Game Studio applications targeting the Xbox 360 console. Avatar support includes retrieving existing gamer avatars or creating new ones, and rendering and animating avatars using built-in support.

Overview

Avatars provide other ways of placing the player in your game. With avatars, you can render and animate the avatar within the game, or you can generate one or more random avatars to use as a crowd backdrop or as additional characters in the game.

You must meet the following requirements before using avatars in your game:

- The game targets the Xbox 360 platform.

Note

Code related to avatars can be present in a Windows game, but all methods will return default values and nothing will render to the screen. This is by design.

- Gamer services must be initiated. For more information, see the Initialization of Gamer Services section in [Gamer Services Overview](#).

The avatar of a signed-in gamer is stored in the [Avatar](#) property. Create one or more random avatars using the [CreateRandom](#) method. This method creates an [AvatarDescription](#) object, initialized with random data. You can then render this avatar with a call to [Draw](#).

XNA Game Studio provides avatar rendering with the [AvatarRenderer](#) class, and animation with the [AvatarAnimation](#) class. The renderer uses basic lighting and shading effects. You can invoke the renderer with a single call to [AvatarRenderer.Draw](#). This method also uses an animation to indicate the avatar currently is being loaded. This is automatically used by the renderer until loading is completed. Once the avatar is displayed, idle animations cycle randomly unless you specify a specific animation from the standard list. For a complete list, see [AvatarAnimationPreset](#). The **AvatarRenderer** class contains properties related to the lighting and rendering of the avatar. These properties include the directional light color and direction, the transformation matrices, and others. For a full listing of these properties, see [AvatarRenderer Properties](#).

See Also

Concepts

Tasks

Reference

[AvatarDescription](#)

[AvatarExpression](#)

[AvatarRenderer](#)

[Avatar](#)

How To: Use Gamertags and Gamer Pictures

Demonstrates how to retrieve gamertags and pictures using a technique that can be applied to retrieve other gamer profile information.

Using Gamertags and Gamer Pictures

To retrieve the gamertag and profile picture of a player in a game

- Use [Gamer.GetProfile](#) to retrieve the profile information for any player in the game.

Note that this method will complete quickly when you use it with a locally signed-in profile. However, this method can take some time if you call it on a remote gamer instance. In the remote case, you might use the non-blocking alternative [Gamer.BeginGetProfile](#).

The [GamerProfile](#) returned by [Gamer.GetProfile](#) also contains such information as the motto, gamerscore, and more.

If the gamer is a [SignedInGamer](#), you can retrieve the local player's preferred settings for gameplay using the [SignedInGamer.GameDefaults](#) property. You can also retrieve the local player's privileges using [SignedInGamer.Privileges](#). Unlike [Gamer.GetProfile](#), which can be slow for remote players, the properties of a [SignedInGamer](#) return instantly.

C#

```
foreach (NetworkGamer gamer in session.AllGamers)
{
    string text = gamer.Gamertag;
    GamerProfile gamerProfile = gamer.GetProfile();
    Texture2D picture = gamerProfile.GamerPicture;
}
```

How To: Add Presence Information

To set presence information for a player, use the [SignedInGamer.Presence](#) property, which is a reference to a [GamerPresence](#) object. **GamerPresence** has two properties that are used to set presence information:

- [PresenceMode](#), which is set to one of the values of [GamerPresenceMode](#).

GamerPresenceMode contains values that represent strings describing what the player is doing in your game. For a full description of each, see the **GamerPresenceMode** reference topic.

- [PresenceValue](#), which is an integer that provides a numeric value used with some of the presence modes such as **Stage**, **Level**, and **Score**.

To set presence information for a player

Set [SignedInGamer.Presence.PresenceMode](#) with one of the values of [GamerPresenceMode](#), as shown:

C#

```
foreach (SignedInGamer signedInGamer in
    SignedInGamer.SignedInGamers)
{
    signedInGamer.Presence.PresenceMode =
        GamerPresenceMode.InCombat;
}
```

Note

Some values will also require [SignedInGamer.Presence.PresenceValue](#) to be set, as described previously.

Note

A game title will not be displayed in a player's presence information on the Xbox LIVE service until that game passes peer review. Once a game passes peer review, the presence information for a player shows three things: a confirmation the player is playing an Xbox LIVE Indie Game, the game's title, and the presence information set by [SignedInGamer.Presence](#). For example:

```
Xbox LIVE Indie Game
My Game Title
Fighting the Boss
```

While a game is in development, but before the game passes peer review, the presence information indicates the developer is using the XNA Creators Club, as shown:

```
XNA Creators Club
creators.xna.com
Fighting the Boss
```

See Also

Reference

[GamerPresence Class](#)

[SignedInGamer.Presence Property](#)

[GamerPresenceMode](#)

How To: Add Support for Game Invitations

The XNA Framework [player match](#) multiplayer network session type enables players to invite friends to play a game online over Xbox LIVE. Games of this type are able to host other players when the session is joinable, when there are open slots, and when game invitations are enabled.

Gamers can send game invitations from the Xbox LIVE friends list or by using the [Guide.ShowGameInvite](#) method. Game Invitations can also be sent to members of a gamer's LIVE Party by using [LocalNetworkGamer.SendPartyInvites](#). For more information about LIVE Parties, see [How To: Add LIVE Party Support](#).

The [NetworkSession.InviteAccepted](#) event receives the notification of an accepted game invitation. For this reason, titles *must* register for the [InviteAccepted](#) event to send game invitations.

Gamers can accept game invitations from the message center or from the friends list. A gamer can also ask to join an existing game session without an invitation by selecting **Join Session In Progress** from the friends list or from the game lobby. Either of these actions will cause an [InviteAccepted](#) event to be sent to the title.

Important

If your title does not register for the [InviteAccepted](#) event, its sessions will always show up as non-joinable.

To register for the [InviteAccepted](#) event

- Register an event handler for the [NetworkSession.InviteAccepted](#) event when your game starts.

You can use the game [constructor](#) or the [Game.Initialize](#) method to register this event.

C#

```
NetworkSession.InviteAccepted +=
    new EventHandler<InviteAcceptedEventArgs>(
        NetworkSession_InviteAccepted);
```

Once the [InviteAccepted](#) event has been registered, the registered event handler will be called when the event is received. This happens either when a game invitation has been accepted or a player has requested to join a session in progress.

To respond to the [InviteAccepted](#) event

- In the handler registered for [InviteAccepted](#), check whether the invitation is for a pre-existing local session by examining [IsCurrentSession](#). If it is not for the current local session, call [NetworkSession.JoinInvited](#) to allow the gamer to join the remote session. If another local gamer has already joined the session, call [NetworkSession.AddLocalGamer](#) to add additional gamers to the session.

C#

```
void NetworkSession_InviteAccepted(object sender,
    InviteAcceptedEventArgs e)
{
    // if this is from a session that other local gamers are already in
    if (e.IsCurrentSession)
    {
        // and that sessions is still running...
        if (session != null)
            session.AddLocalGamer(e.Gamer); // add the additional gamer
    }
    else // else the invite is to a different session
    {
        if (session != null)
        {
            session.Dispose();
            session = null;
        }
        session = NetworkSession.JoinInvited(maxLocalGamers);
    }
}
```

```
}
```

Note

If an invitation is pending already because the game started in response to a cross-title invite, the method to handle the game invitation event will be called as soon as you subscribe to the event. Because of this, it is important to initialize everything upon which the event handler depends *before* it is registered.

When a game is able to host other gamers, the **Send Game Invite** option button automatically appears in the Xbox LIVE friends list. Because a gamer can send invitations at any point in the game by pressing the **Guide** button and navigating to the friends list, typically you won't need to write additional code to display the game invitation screen.

However, there may be situations when you'd like to bring up the game invitation UI yourself as a result of player actions in the game. In this case, you can use the [Guide.ShowGameInvite](#) method in your title's code to bring up the UI directly.

To display the game invitation user interface

- Call [Guide.ShowGameInvite](#) to show the game invitation UI.

You can do this from any point in your code, but be sure that the title already is registered to receive the [InviteAccepted](#) event as described above.

C#

```
if (IsButtonPressed(GamePadButton.Y) && !Guide.IsVisible)
{
    Guide.ShowGameInvite(PlayerIndex.One, null);
}
```

Invitations to Games that Have Not Been Installed

When a player accepts an invitation for an Xbox LIVE Indie Game that is not already installed on his or her console, the invited player will be taken to the game offer. The offer enables the invited player to download the game and accept the invitation to start the game.

For Creator's Club games still waiting to pass peer review and for which no game offer exists, a game invitation will still be sent over the Xbox LIVE service to the other player. If the game is not installed, the invited player will be prompted with a message indicating that he or she can get the game from the game creator.

See Also

Tasks

[How To: Add LIVE Party Support](#)

Reference

[NetworkSession.AllowJoinInProgress Property](#)

[Guide.ShowGameInvite Method](#)

[NetworkSession.InviteAccepted Event](#)

[NetworkSession.JoinInvited Method](#)

[NetworkSession.BeginJoinInvited Method](#)

[NetworkSession.EndJoinInvited Method](#)

How To: Add LIVE Party Support

LIVE Party gives groups of gamers additional ways to connect, chat, and play together. The XNA Framework supports LIVE Party through the [Guide](#), [LocalNetworkGamer](#), and [SignedInGamer](#) classes. This topic shows you how to use these classes to add LIVE Party support to your game title.

Note

Currently, the only platform that supports LIVE Party is the Xbox 360. Other platforms do not support LIVE Party functionality, even though the platform may support LIVE otherwise.

Showing the LIVE Party Screens

The Guide provides two screens related to LIVE Party: the Party screen and Party Sessions screen.

The Party screen serves as the main interface to LIVE Party functionality for a given player. It allows a player to do the following:

- View all the members of the current party.
- Manage the party (if the player is the party organizer).
- Send party invitations to other party members.

The Party Sessions screen allows a player to see all party members who are in a joinable session in the same game. It allows the player to join any of these sessions.

To show the Guide's LIVE Party screen

- Call [Guide.ShowParty](#) with the [PlayerIndex](#) of the player for which the Party Sessions screen will be shown.

C#

```
// Show the party screen for the player on the first controller.
Guide.ShowParty(PlayerIndex.One);
```

To show the Guide's LIVE Party Sessions screen

- Call [Guide.ShowPartySessions](#) with the [PlayerIndex](#) of the player for which the Party Sessions screen will be shown.

C#

```
// Show the party sessions screen for the player on the first controller.
Guide.ShowPartySessions(PlayerIndex.One);
```

Sending a Game Invitation to Party Members

For party members not yet in the game, the player can use the [LocalNetworkGamer](#) class to send them an invitation to the current game session.

To send a game invitation to all party members not in the current game session

- Call [LocalNetworkGamer.SendPartyInvites](#) for the gamer for whom invitations will be sent.

C#

```
// Send invitations to all members of each local network gamer's LIVE party.
foreach(LocalNetworkGamer gamer in NetworkSession.LocalGamers)
{
    gamer.SendPartyInvites();
}
```

Note

When a gamer responds to a game invitation, the person who sent the invitation gets the [NetworkSession.InviteAccepted](#) event. For more information, see [How To: Add Support for Game Invitations](#)

Retrieving the Number of Party Members for a Gamer

You can use the gamer's [SignedInGamer](#) class to determine the number of party members in a gamer party.

To get the number of party members for a player.

- Read the value of [SignedInGamer.PartySize](#).

Note

[PartySize](#) is read-only. It cannot be used to set the party size.

C#

```
// get the party size for the first signed-in gamer.
if(Gamer.SignedInGamers(0).PartySize > 0)
{
    // The player is a member of a LIVE party!
    ...
}
```

See Also

Tasks

[How To: Add Support for Game Invitations](#)

Reference

[Guide.ShowParty](#) Method

[Guide.ShowPartySessions](#) Method

[SignedInGamer.PartySize](#) Property

[LocalNetworkGamer.SendPartyInvites](#) Method

How To: Initialize and Update the Gamer Services Dispatcher

Describes how an application would use the [GamerServicesDispatcher](#) instead of the [GamerServicesComponent](#) for cases in which a program may not be using the XNA Framework application model or component infrastructure.

Automatically Updating the Gamer Services Dispatcher

The XNA Framework provides a [GamerServicesComponent](#). This is a game component that automatically takes care of initializing and updating the gamer services dispatcher. To make use of this component, XNA Framework games need to add only one line of code to the [Game](#) constructor.

To automatically update the gamer services dispatcher

- To make use of this component, add this line of code to the [Game](#) constructor:

C#

```
Components.Add(new GamerServicesComponent(this));
```

Manually Updating the Gamer Services System

In some cases a program might not use the XNA Framework application model or component infrastructure. For this application, it is possible to call the [GamerServicesDispatcher](#) directly.

To manually initialize and update the gamer services system

- Once, in the startup code for the application, call [GamerServicesDispatcher.Initialize](#) to initialize the gamer services subsystem.

C#

```
protected override void Initialize()
{
    GamerServicesDispatcher.WindowHandle = Window.Handle;

    GamerServicesDispatcher.Initialize(Services);

    base.Initialize();
}
```

- Call [GamerServicesDispatcher.Update](#) once every frame to update the gamer services system.

C#

```
protected override void Update(GameTime gameTime)
{
    GamerServicesDispatcher.Update();
    base.Update(gameTime);
}
```

See Also

Concepts

[Gamer Services Overview](#)

How To: Detect or Simulate Trial Mode and Present a Marketplace Offer

Demonstrates how to detect and simulate trial mode for the purpose of testing the display of a Marketplace offer that will allow a player to purchase your game.

Note

It is not necessary to write special code to ensure that a Marketplace offer is available to players of Xbox LIVE Indie Games. By default, trial mode games that have been released will present a Marketplace offer to the player when the gameplay time limit expires. However, you might want to write special code in your game to present a Marketplace offer, or control other areas of gameplay when a game is in trial mode.

To simulate trial mode

The XNA Framework provides a property, [Guide.SimulateTrialMode](#), that allows developers to test the behaviors of a game under trial mode conditions. When games are in trial mode, a developer might choose to limit the capabilities of the game.

- In this example, [Guide.SimulateTrialMode](#) is set to **true** for the purpose of later testing a marketplace offer that is presented if the game is in trial mode. This property is usually set in the game [constructor](#), and must be set before the first call to [Game.Update](#).

C#

```
Guide.SimulateTrialMode = true;
```

To detect when a game is in trial mode

- Check whether [Guide.IsTrialMode](#) is **true**.
- In this example, if this condition is met, call [Guide.ShowMarketplace](#) to present an offer to purchase the full version of the game.

C#

```
if (Guide.IsTrialMode)
{
    Guide.ShowMarketplace(signedInGamer.PlayerIndex);
}
```

How To: Render and Animate an Avatar Using AvatarRenderer.

Demonstrates how to render and animate a gamer's avatar using the [AvatarRenderer](#) class and a standard animation.

Note

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen. This is by design.

The Complete Sample

The code in this topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download AnimatedAvatar_Sample.zip](#).

To Render a Random Model with a Standard Animation

1. Initialize the gamer services dispatcher for your application.

C#

```
Components.Add(new GamerServicesComponent(this));
```

For more information, see [How To: Initialize and Update the Gamer Services Dispatcher](#).

2. Declare three new members in your main game class to store the various objects needed to render and animate the random avatar.

C#

```
AvatarDescription avatarDesc;
AvatarRenderer avatarRenderer;
AvatarAnimation avatarAnimation;
```

3. Create and initialize the avatar data to a random set with a call to [CreateRandom](#). Use the [Initialize](#) method of your [Game](#) class, after the existing call to [Initialize](#).

C#

```
avatarDesc = AvatarDescription.CreateRandom();
```

4. Create a new avatar renderer for the recently-created random avatar, with the standard loading effect. Use the [Initialize](#) method of your [Game](#) class.

C#

```
avatarRenderer = new AvatarRenderer(avatarDesc, true);
```

5. Create a new [AvatarAnimation](#) object, and load any standard animation from the list supported by [AvatarAnimationPreset](#). Use the [Initialize](#) method of your [Game](#) class.

For this example, because the avatar gender is unknown at the time of rendering, use the gender-neutral clap animation.

Note

Gender-specific animations work with any avatar, but each was designed with the proportions of that gender in mind. This prevents the avatar model from clipping through itself during the animation cycle.

C#

```
avatarAnimation = new AvatarAnimation(AvatarAnimationPreset.Clap);
```

6. Initialize the avatar's transformational matrices (world, projection, and view), using the [World](#), [Projection](#), and [View](#) properties of the `avatar` object. Use the [Initialize](#) method of your [Game](#) class.

C#

```
avatarRenderer.World =  
    Matrix.CreateRotationY(MathHelper.ToRadians(180.0f));  
avatarRenderer.Projection =  
    Matrix.CreatePerspectiveFieldOfView(MathHelper.ToRadians(45.0f),  
    GraphicsDevice.Viewport.AspectRatio, .01f, 200.0f);  
avatarRenderer.View =  
    Matrix.CreateLookAt(new Vector3(0, 1, 3), new Vector3(0, 1, 0),  
    Vector3.Up);
```

Note

The values used in this example are for demonstration purposes only. Although they provide a good location and orientation for the camera (in terms of this sample), your application will require different values, which are determined by your application's needs.

7. Make any necessary modifications to the current values of the avatar's ambient light and directional light properties, using the similarly-named properties of the `avatar` object (for example, `AvatarRenderer.AmbientLightColor`).
8. Update the current animation of the avatar.

The recommended place for this code is within the `Update` method of your application.

C#

```
if (avatarRenderer.IsLoaded)  
{  
    avatarAnimation.Update(gameTime.ElapsedGameTime, true);  
}
```

9. In the `Draw` method of your game, render the avatar by calling `Draw`.

C#

```
avatarRenderer.Draw(avatarAnimation.BoneTransforms,  
    avatarAnimation.Expression);
```

When you render the avatar, keep in mind that various parts of the avatar model can be transparent (for example, glasses). Any transparent parts are rendered with the depth buffer disabled. Therefore, first render all opaque objects in the scene, except the avatar. Then render the avatar and any other transparent objects, sorted by depth (from back to front).

In addition, avatar rendering ignores the current state of the `RenderState.FillMode` property. It is always treated as if `FillMode.Solid` is the current value.

See Also [Programming with Avatars](#)

Networking

Overviews

[Getting Started With Networked Games](#)

Introduces the key prerequisites for creating a networked XNA Framework game.

[Network Session Management](#)

Defines the network session management capabilities in the XNA Framework, which allow for matchmaking access to profile data on all the gamers in a potential game.

[Reliable Packet Delivery](#)

Provides an overview of the reliable UDP protocol supported by the XNA framework and the packet delivery options that may be chosen.

[Network Topologies and Host Migration](#)

Discusses the relationship between session hosts and peers, and the actual network topology used to route gameplay.

[Latency Simulation and Bandwidth Usage](#)

Describes the packet loss simulation capabilities available to XNA Framework games.

[Voice Support](#)

Introduces the voice support options available in the XNA Framework.

Networking How-Tos

[How To: Create a Network Session](#)

Describes the process of creating a new network session.

[How To: Find and Join a Network Session](#)

Describes how to find and join a network session.

[How To: Manage Players Joining and Leaving the Game](#)

Manages the data associated with a player by subscribing to events that occur when players are joining or leaving a game.

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Describes the properties and events associated with a multiplayer session that allow a game to move between lobby and game play modes. It identifies the game's current mode.

[How To: Join an In-Progress Game](#)

Describes how to enable the [AllowJoinInProgress](#) property to make an in-progress game available to peers searching for available sessions.

[How To: Send Data](#)

Describes the options available for sending data to all clients or to a specific player.

[How To: Receive Data](#)

Describes how the local players on a gaming machine receive data from the other players in a networked game.

See Also

Concepts

[Gamer Services Overview](#)

[Zune Networking Overview](#)

[Networking Content Catalog at XNA Creators Club Online](#)

Tasks

[Third-Party Firewall Settings](#)

[How To: Build a Simple Networked Game on Zune](#)

[How To: Add Presence Information](#)

[How To: Add Support for Game Invitations](#)

[How To: Initialize and Update the Gamer Services Dispatcher](#)

[How To: Use Gamertags and Gamer Pictures](#)

[How To: Detect or Simulate Trial Mode and Present a Marketplace Offer](#)

Reference

[Microsoft.Xna.Framework.Net Namespace](#)

Getting Started With Networked Games

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

Dependency on Gamer Services

XNA Framework games must explicitly initialize the *gamer services* system before networking methods are called. Gamer services allows your game to receive notifications such as sign-in status changes, game invitations and other messages from Xbox LIVE, and it gives your game access to the Guide. The Guide provides a user interface allowing players to sign in to Xbox LIVE, view messages or game invitations, and use Xbox LIVE user interface elements such as the on-screen keyboard during a game.

Most games use [GamerServicesComponent](#), a game component that wraps the [GamerServicesDispatcher](#) object and takes care of sending and receiving messages from Xbox LIVE. This component also passes in the graphics device and window handle to allow rendering of the Guide, and it calls [GamerServicesDispatcher.Update](#) at regular intervals. To use this component, most games only need to add one line to the [Game](#) constructor:

C#

```
Components.Add(new GamerServicesComponent(this));
```

For more information on Gamer Services, see [Gamer Services Overview](#).

XNA Creators Club and Xbox LIVE Membership Requirements

There are two types of online multiplayer games available when you create a networked game with XNA Game Studio: system link game sessions and LIVE sessions. These two forms of online multiplayer gaming describe whether the multiplayer gaming session is on the local area network (system link games), or if the gaming session is advertised and found through the LIVE service. With system link games, you can search the local area network to find the gaming sessions. With LIVE sessions, you can use the LIVE service to find available game sessions. Use the *searchProperties* argument of [NetworkSession.Find](#) to specify the parameters used to search for an available game session—a game session that uses either system or the LIVE service.

Developing and testing a networked game requires at least two machines, but you only need one Xbox 360 console and one Creators Club membership to test network code for the Xbox 360. This is because XNA Framework supports cross-platform system links so developers can run one instance of a game on an Xbox 360, and a second on a Windows-based computer. This functionality helps creators debug their titles without having to purchase a second console. It is also possible for more than one Windows-based development computer to connect several machines in a system link session without any memberships required.

Memberships are required for a player on a Windows-based computer and an Xbox 360 console in a network session, depending on the session type that has been created.

	Xbox 360 console	Windows-based development computer	Zune
Run an XNA Framework Game	LIVE Silver membership + Premium XNA Creators Club membership	No memberships required	No memberships required
Use System Link for Local Area Network gameplay	LIVE Silver membership + Premium XNA Creators Club membership	No memberships required	No memberships required
Sign in to Xbox LIVE and Games for Windows - LIVE Servers	LIVE Silver membership + Premium XNA Creators Club membership	LIVE Silver membership + Premium XNA Creators Club membership	Not available on Zune
Use LIVE to connect to other machines over the Internet while the game is in development	LIVE Gold membership + Premium XNA Creators Club membership	LIVE Silver membership + Premium XNA Creators Club membership	Not available on Zune

Caution

XNA Framework components that interact with Games for Windows - LIVE require an XNA Creators Club membership. If for any reason a LIVE profile without a Creators Club membership is set to auto sign in to a Windows-based computer, the game will be unable to connect to the LIVE service. Furthermore, the XNA Framework gamer services components, including the Guide, will be unavailable. To disable auto sign in for a LIVE profile on a Windows-based computer, disconnect from the network, launch an XNA Framework game, and then open the Guide.

Hardware Requirements

It is not possible to run multiple simultaneous instances of a networked game on a single development computer.

When you launch to multiple machines for network testing, we recommend that you load multiple copies of your solution into separate Visual Studio instances, one per machine.

For a system link game, you can use any one of the following hardware configurations to test a networked game:

- *One development computer and one Xbox 360 console* (running one instance of the game on a Windows-based computer and one on Xbox 360). The creator needs one LIVE Silver and one Creators Club membership to run code on the Xbox 360.
- *One development computer and one client computer* (running one instance of the game on each Windows-based computer). Creators cannot run two networked XNA Framework games at the same time on the same computer. Creators need a second machine to run a second game instance for testing purposes. No LIVE memberships or Creators Club memberships are required for this scenario.
- *One development computer and two Xbox 360 consoles* (running one instance of the game on each Xbox 360). Here a creator deploys and debugs the game on two Xbox 360s. The creator needs at least two LIVE Silver memberships and two Creators Club memberships (one pair for each Xbox 360) for this scenario.
- *One development computer and two Zune devices* (running one instance of the game on each Zune). Here a creator deploys and debugs the game on two Zune devices. The creator needs no additional memberships for this scenario. Note that Zune only supports system link networking and does not support cross-platform networking or internet connectivity.

If a creator wants to use LIVE to connect to gaming machines over the Internet as opposed to using system link to connect gaming machines over a local subnet, you may use any one of the following hardware configurations:

- *One development computer and one Xbox 360 console* (running one instance of the game on the development computer and one on the Xbox 360). The creator needs two LIVE Gold memberships and two Creators Club memberships (one pair for each machine) for this scenario.
- *One development computer and one client computer* (running one instance of the game on each computer). The creator needs two LIVE Gold memberships and two Creators Club memberships (one pair for each machine).
- *One development computer and two Xbox 360 consoles* (running one instance of the game on each Xbox 360 console). The creator needs two LIVE Gold memberships and two Creators Club memberships (one set for each machine).

See Also

Concepts

[Zune Networking Overview](#)

Tasks

[Third-Party Firewall Settings](#)

Network Session Management

Any matchmaking scenario involves the creation of a network session. Network sessions give XNA Framework games access to profile data on all the gamers in a potential game.

- [Session Updates](#)
- [Matchmaking](#)
- [Session Types](#)
- [Finding Sessions](#)
- [Session Management](#)

Session Updates

The [NetworkSession](#) class has a single [Update](#) method that must be called once per frame. This update call performs the following actions:

- Sends the network packets.
- Changes the session state, such as which players are in the session and which player is talking currently.
- Raises the managed events for any significant state changes.
- Returns the incoming packet data.

This explicit update call allows the framework to synchronize the session so that packet-received and state-change events will never be raised asynchronously with the rest of the game code. This allows developers to program against the network session object without any threading concerns.

Session updates are kept separate from the gamer services system pumping for two reasons:

- To clarify when session events will be raised.
- To allow developers the option to update the session from a background worker thread, which allows these developers the option to run this in parallel with the main game update.

Matchmaking

To play a networked game with other people, the game needs a way to publish an instance of the game for others to discover the instance and join the game. This matchmaking and peer discovery mechanism is available through the [NetworkSession.Find](#) and [NetworkSession.BeginFind](#) methods, which return a collection of [AvailableNetworkSession](#) instances. You may examine the properties of each [AvailableNetworkSession](#), and pass a suitable [AvailableNetworkSession](#) to [Join](#) to join the session.

Players

The [NetworkSession](#) object exposes a collection of player instances, along with [events](#) indicating when players join or leave the session. There are properties for querying which player is the host ([NetworkSession.Host](#)), and for determining which players are local players signed on to the same machine and physically located together ([NetworkGamer.IsLocal](#)).

Friends

The XNA Framework exposes basic functionality for programmatic friends list access:

- Query the contents of the friends list using [SignedInGamer.GetFriends](#), obtaining gamertags and presence information using the methods and properties available from a [FriendGamer](#).
- Query whether another player is a friend using [SignedInGamer.IsFriend](#).

Local Sign-In

When you first create a session, a game must specify the maximum number of local players that it supports (1, 2, 3, or 4). The framework will automatically add up to this number of profiles to the session, and keep this up to date if the sign-in state changes. If there are more signed-in profiles than the game wants, only the lower-numbered profiles will be added to the session. If extra players sign after the session is created the players will not be added. If the host player signs off, the session will either be torn down or a host migration will take place depending on whether the [NetworkSession.AllowHostMigration](#) option is enabled.

Session Types

The XNA Framework supports four kinds of sessions:

- Local
- System link
- Xbox LIVE unranked session
- LIVE ranked session

The same matchmaking API is used for both LIVE and system link session types. The session type must be specified when creating or searching for sessions, and only sessions of an identical type will be returned by the query.

Local sessions are interesting for games that want to build both networking and local split-screen multiplayer on top of the same infrastructure. This allows this type of game to implement both online and offline game modes using the same framework session interfaces.

Finding Sessions

The XNA Framework provides a single fixed matchmaking query in the form of a [NetworkSessionProperties](#) collection that represents the desired session to match. This collection is passed to the [NetworkSession.Find](#) and [NetworkSession.BeginFind](#) static methods. These methods return an [AvailableNetworkSessionCollection](#) containing a maximum of 25 [AvailableNetworkSession](#) objects, which in turn contain the following data about each session:

- [Gamertag of the host](#).
- [Number of players in the network session](#).
- Number of free [public](#) and [private](#) slots. On Windows- and Xbox 360-based games, the supported range is between 2 and 31. On the Zune platform, the supported range is between 2 and 8.
- [Quality of service](#) information. This information is filled in over time, so search results will be returned initially without quality of service data, that will then become available at a later time.

Session Management

One important task in writing a networked game is managing the session, or instance, of the game. The events on the [NetworkSession](#) object allow you to be aware of the following:

- A player connects or disconnects using [GamerJoined](#) and [GamerLeft](#)
- The game has changed state between lobby, gameplay, and post-gameplay modes using [GameStarted](#), [GameEnded](#), and [SessionEnded](#)
- The network machine hosting the session changes using [HostChanged](#)

Lobby Transitions

The XNA Framework keeps track of whether the session is in the lobby or actually playing the game. It enables events to notify titles when the state changes. While in the lobby, each player can use [IsReady](#) to signal that he or she is ready to move from the lobby and into the game. The host can check [IsEveryoneReady](#) to see if all the players are ready to begin the game.

Each network session can use [AllowJoinInProgress](#) to specify whether the game supports join-in-progress. This is used in conjunction with the lobby versus play state to automatically update the session joinability status on the matchmaking servers.

See Also

Tasks

[How To: Create a Network Session](#)

[How To: Manage Players Joining and Leaving the Game](#)

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Reliable Packet Delivery

Reliable TCP sockets generally are considered unsuitable for game networking, for a variety of reasons. Most of the reasons have to do with how the sockets handle congestion and how they prioritize in-order delivery over timeliness. Most games use UDP sockets, which means they send the vast bulk of their gameplay data unreliably. Reliable delivery is still necessary for some subsets of the data, however. To improve reliability, the XNA Framework provides a reliable delivery mechanism on top of UDP.

The XNA Framework [SendData](#) and [ReceiveData](#) methods are built over a reliable UDP packet layer. This packet layer provides optional ordering and/or delivery guarantees, plus automatic packet coalescence (if several packets are sent in rapid succession) and splitting (if the payload is too big for a single UDP wire packet) to make efficient use of the underlying UDP channel.

Fundamentally, this is a peer-to-peer architecture with reliable connections from each machine to all other machines. However, this does not place any limitation on what topologies the game can use for its own network traffic. Client-server, peer-to-peer, or any kind of hybrid architecture can benefit from the reliable UDP protocol supported by the XNA Framework.

Packet Delivery Options and Network Performance

There are five packet delivery options available through the [SendDataOptions](#) parameter that is passed to any call to [SendData](#):

- Unreliable, out-of-order ([SendDataOptions.None](#))
- Unreliable, in-order ([SendDataOptions.InOrder](#))
- Reliable, out-of-order ([SendDataOptions.Reliable](#))
- Reliable, in-order ([SendDataOptions.ReliableInOrder](#))
- Chat data ([SendDataOptions.Chat](#))

If a reliable sequential message sent using [SendDataOptions.ReliableInOrder](#) is dropped, later sequential messages cannot be delivered until the message is resent and received. In effect, this stalls the channel. If you use this option to send more messages, you are going to see real-world bottlenecks. If the receiver does not call [ReceiveData](#) quickly enough or just stops altogether, the queue of messages to resend fills up.

Unreliable sequential messages sent using [SendDataOptions.InOrder](#) may have gaps in the sequence, but they will never go backward or stall because of previous non-reliable sequential messages. Unreliably sent data will not be resent. This is good for information that is updated frequently, information such as player position. Newer information will be available by the time the sender attempts to resend the data. For all game data that is sent regardless of the delivery option, always be sure to send the minimum amount of data necessary to communicate the information. This can be done by sending only changes in the game state rather than the entire game state.

For best performance in a network game, minimize the use of reliable and in-order packet delivery. In general, reliable packet delivery should only be used for data critical for your game to function properly, as it consumes bandwidth resending packets as necessary.

[SendDataOptions.Chat](#) can be included independently of any other flags, and has two effects. First, any in-order chat sends are only guaranteed to be ordered with regard to other chat sends. There is no guarantee of ordering between non-chat and chat sends. Second, data within a chat send is not encrypted between the sender and receiver.

See Also

Tasks

[Third-Party Firewall Settings](#)

Network Topologies and Host Migration

Although the XNA Framework specifies one machine as the host of a session, this does not imply anything about the network topology used to route the actual gameplay. The host is responsible for the following:

- Owning the session description on the Xbox LIVE servers
- Updating multiplayer session properties as the game state changes
- Controlling when to transition between the lobby and play states
- Removing players from the session (peers cannot directly remove other peers – only the host can do this)

Games are free to choose any network topology for their gameplay data:

- Peer-to-peer
- Client-server, where the host doubles as the server
- Client-server, using one of the other session members as the server
- Any other hybrid approach

The host and server are often the same, but they are not required to be. The host has significance to the LIVE service, while the server relates purely to gameplay. The host is chosen by the XNA Framework. The host will be the machine that creates the session. However, the host may change if there is a host migration. The game architecture determines which server is chosen. The host is an important part of XNA Framework gamer services, but the concept of a server only exists inside the game code. There are no types or properties relating to servers in the framework itself. Games need not have any server at all if they are using a pure peer-to-peer architecture.

Hosts and Peers

One of the gaming machines is designated the host for a LIVE multiplayer session. The other gaming machines are peers. The concept of host or peer is independent of the network topology chosen for the game. That is, the concept of host does not necessarily imply that the console is the network host, or even that a client-host topology is in use, even though many titles align the two concepts.

The XNA Framework automatically handles host migration. When a host leaves, the run time will automatically elect a new host. It then raises an event to inform the game of the change. Support for host migration is optional. It is disabled by default.

Note that the XNA Framework automatically migrates the host, but not the server. The game is still responsible for migrating the simulation state. This is often trivial for peer-to-peer titles, but can be more challenging for client-server architectures. If the host quits the game gracefully through a menu selection, the game code could explicitly transfer the server simulation state before actually leaving the session. However, this is very difficult to do, particularly for the more general case of forced exits such as a hardware reboot.

Latency Simulation and Bandwidth Usage

To debug and test a networked game, you need to create at least two peers and have them communicate. There are several ways to create multiple peers, including running multiple instances across equal numbers of computers or a mix of computers and Xbox 360 consoles. Most development will take place on a local area network (LAN) where network throughput and quality is not an issue. Networked games that run across the Internet, however, will need to handle a number of issues such as lower bandwidth availability, non-reliable delivery of messages, and firewall and proxy support.

To help developers test networked games over a subnet where you will typically see much lower levels of latency and packet loss than can be expected, the XNA Framework includes a latency and packet loss simulator. This simulator requires no special hardware, extra network cards, or unusual Windows network configurations. You can use the latency simulation capabilities to test techniques, such as client-side prediction, that you might use in your game design.

You can use [NetworkSession.SimulatedLatency](#) to set the desired latency, and you can use [NetworkSession.SimulatedPacketLoss](#) to set a desired packet loss percentage.

The network simulation respects the [SendDataOptions.Reliable](#) and [SendDataOptions.InOrder](#) flags. Packets that are sent reliably will never be discarded, no matter how high [SimulatedPacketLoss](#) is set. Packets sent without [SendDataOptions.InOrder](#) will be delayed by a random value from a bell curve centered on the current [SimulatedLatency](#) setting, which has the effect of reordering packets in addition to delaying them. Packets that use [SendDataOptions.InOrder](#) will be delayed without any reordering.

For a game to play well over the Internet, it is recommended that you design it to handle up to 0.1 (10%) packet loss, and give it 200 milliseconds of latency.

The XNA Framework also includes performance metrics with the [NetworkSession.BytesPerSecondSent](#) and [NetworkSession.BytesPerSecondReceived](#) so games can easily measure upstream and downstream bandwidth usage.

Voice Support

Voice data is automatically transmitted and replayed without any title effort whatsoever. The [GamerPrivileges.AllowCommunication](#) property enables you to see which players are allowed to send voice data. The [LocalNetworkGamer.EnableSendVoice](#) method allows a game to programmatically specify which players can talk. Titles can use this to implement team chat or proximity voice. The XNA Framework does not expose direct program access to the voice data stream.

There are three Boolean properties exposed to query the voice status of another player:

- Do they have voice access? [NetworkGamer.HasVoice](#)
- Are they currently talking? [NetworkGamer.IsTalking](#)
- Are they muted? [NetworkGamer.IsMutedByLocalUser](#)

These properties allow games to display voice-related information in the user interface. One example would be flashing the name of the player who is currently talking.

How To: Create a Network Session

- [Conducting a Network Session](#)

A network session is made up of the players in the game and an optional set of attributes to aid in describing the type of session that is being created. The attributes that are used to describe an available multiplayer session are known as session properties. These session properties are created by you to describe this particular type of multiplayer session for your particular game. They might contain descriptions of details such as a whether this game is a timed session or capture-the-flag session.

Note

Creating a [NetworkSessionProperties](#) collection to describe your session is optional. You can also pass in **null** when creating a new network session.

The [NetworkSessionProperties](#) supports up to eight integer values that describe the session. Because these values are integers, we will create a set of enums in this example to describe these integer values and their placement in the [NetworkSessionProperties](#).

C#

```
enum SessionProperty{ GameMode, SkillLevel, ScoreToWin }

enum GameMode{ Practice, Timed, CaptureTheFlag }

enum SkillLevel{ Beginner, Intermediate, Advanced }
```

Conducting a Network Session

To describe session properties

1. Create a new [NetworkSessionProperties](#).
2. Set the values to the custom session properties.

C#

```
NetworkSessionProperties sessionProperties =
    new NetworkSessionProperties();

sessionProperties[(int)SessionProperty.GameMode]
    = (int)GameMode.Practice;
sessionProperties[(int)SessionProperty.SkillLevel]
    = (int)SkillLevel.Beginner;
sessionProperties[(int)SessionProperty.ScoreToWin]
    = 100;
```

You can create many different types of sessions. One is a local session, which is used for split-screen gaming and requires no network traffic. Another is a system link session, which connects multiple gaming machines over a local subnet. Finally, there is an Xbox LIVE multiplayer session, which takes place over the Internet. In this example, we will create a system link game.

Note

The gaming machine that creates the session is called the host. It owns the multiplayer session. The host does not imply a game server or authority. There are no network topologies implied, such as client-server, peer-to-peer, or hybrid. The network topology will be determined by your implementation of your game.

C#

```
NetworkSession session;

int maximumGamers = 8;
int privateGamerSlots = 2;
int maximumLocalPlayers = 1;
```



```
// Create the session
session = NetworkSession.Create(
    NetworkSessionType.SystemLink,
    maximumLocalPlayers, maximumGamers, privateGamerSlots,
    sessionProperties );
```

To enable host migration and join-in-progress functionality

- Set the [AllowJoinInProgress](#) and [AllowHostMigration](#) properties to values appropriate for your game.

C#

```
session.AllowHostMigration = true;
session.AllowJoinInProgress = true;
```

To subscribe to session events

- Subscribe to any session events in which your game has an interest.

Multiplayer sessions have events that occur when the game transitions from lobby to gameplay ([GameStarted](#) and [GameEnded](#)), when players join and leave ([GamerJoined](#) and [GamerLeft](#)), when the host changes ([HostChanged](#)), and when the session ends ([SessionEnded](#)).

C#

```
session.GamerJoined +=
    new EventHandler<GamerJoinedEventArgs>(session_GamerJoined);
session.GamerLeft +=
    new EventHandler<GamerLeftEventArgs>(session_GamerLeft);
session.GameStarted +=
    new EventHandler<GameStartedEventArgs>(session_GameStarted);
session.GameEnded +=
    new EventHandler<GameEndedEventArgs>(session_GameEnded);
session.SessionEnded +=
    new EventHandler<NetworkSessionEndedEventArgs>(
        session_SessionEnded);
```

To end the session

- If you are a host ending the session purposefully and the conditions for ending the session have been met, call [NetworkSession.EndGame](#).

In this example, when the host presses the ESC key or **Back** button indicating that the player wants to exit the game, the host checks to see whether there are players still in the game. If not, the host ends the session.

C#

```
// Check for exit.
if ( IsButtonPressed(GamePadButton.B ) ||
    IsButtonPressed( GamePadButton.Back ) )
{
    if (session != null)
    {
        if (session.AllGamers.Count == 1)
        {
            session.EndGame();
            session.Update();
        }
    }
}
```

The end of a game is controlled by the [NetworkSession.EndGame](#) method. At the end of a game, you might choose to move back to the lobby or to end the session. Players in the game can subscribe to the [NetworkSession.GameEnded](#) and

[NetworkSession.SessionEnded](#) events.

See Also

Concepts

[Network Session Management](#)

[Third-Party Firewall Settings](#)

Tasks

[How To: Find and Join a Network Session](#)

How To: Find and Join a Network Session

Describes how to find and join a network session.

The [NetworkSessionProperties](#) collection used to create a session with specific properties can also be used to search for sessions with particular properties. Because these values are integers, we will create a set of enumerations in this example to describe these integer values and their placement in the [NetworkSessionProperties](#) collection.

Note

Setting session properties to use as search parameters is an optional step. You may also pass in an empty [NetworkSessionProperties](#) to [NetworkSession.Find](#) or [NetworkSession.BeginFind](#) to view all available sessions regardless of property settings.

C#

```
enum SessionProperty{ GameMode, SkillLevel, ScoreToWin }

enum GameMode{ Practice, Timed, CaptureTheFlag }

enum SkillLevel{ Beginner, Intermediate, Advanced }
```

Joining a Network Session

To set the search parameters

1. Create a [NetworkSessionProperties](#) collection.
2. Set the values to the desired search parameters.

C#

```
NetworkSessionProperties searchProperties =
    new NetworkSessionProperties();
searchProperties[(int)SessionProperty.GameMode] =
    (int)GameMode.Practice;
searchProperties[(int)SessionProperty.SkillLevel] =
    (int)SkillLevel.Beginner;
```

To find an available network session

1. Create an [AvailableNetworkSessionCollection](#) to hold the search results.

C#

```
AvailableNetworkSessionCollection availableSessions;
```

2. Use [NetworkSession.Find](#) to retrieve the list of available sessions.
3. Specify the type of session in which you are interested, the number of local players that wish to join, and any session properties that should be matched when searching.

Note that the *searchProperties* argument is optional. You may pass in **null** to match any available session regardless of the session properties settings for that session.

C#

```
int maximumLocalPlayers = 1;
availableSessions = NetworkSession.Find(
    NetworkSessionType.SystemLink, maximumLocalPlayers,
    searchProperties);
```

Note

This search may also be performed asynchronously using [NetworkSession.BeginFind](#) and [NetworkSession.EndFind](#).

Once you have a list of available sessions, you can examine the session to determine if you would like to join a particular session. In this example, we will print some information about the available session for the player to examine.

C#

```
int sessionIndex = 0;
AvailableNetworkSession availableSession =
    availableSessions[sessionIndex];

string HostGamerTag = availableSession.HostGamertag;
int GamersInSession = availableSession.CurrentGamerCount;
int OpenPrivateGamerSlots =
    availableSession.OpenPrivateGamerSlots;
int OpenPublicGamerSlots =
    availableSession.OpenPublicGamerSlots;
string sessionInformation =
    "Session available from gamertag " + HostGamerTag +
    "\n" + GamersInSession +
    " players already in this session. \n" +
    +OpenPrivateGamerSlots +
    " open private player slots available. \n" +
    +OpenPublicGamerSlots + " public player slots available.";

spriteBatch.DrawString(spriteFont, sessionInformation,
    new Vector2(100, y), color);
```

To join an available multiplayer session

1. Create a [NetworkSession](#).

C#

```
NetworkSession session;
```

2. Join the [AvailableNetworkSession](#) using [NetworkSession.Join](#).

C#

```
session = NetworkSession.Join(
    availableSessions[selectedSessionIndex]);
```

3. After joining a [NetworkSession](#), subscribe to any session events that the peer might be interested in.

C#

```
session.GamerJoined +=
    new EventHandler<GamerJoinedEventArgs>(session_GamerJoined);
session.GamerLeft +=
    new EventHandler<GamerLeftEventArgs>(session_GamerLeft);
session.GameStarted +=
    new EventHandler<GameStartedEventArgs>(session_GameStarted);
session.GameEnded +=
    new EventHandler<GameEndedEventArgs>(session_GameEnded);
session.SessionEnded +=
    new EventHandler<NetworkSessionEndedEventArgs>(
    session_SessionEnded);
```

See Also

Concepts

[Network Session Management](#)

Tasks

[How To: Create a Network Session](#)

How To: Manage Players Joining and Leaving the Game

Manages the data associated with a player by subscribing to events that occur when players are joining or leaving a game.

A [NetworkSession](#) contains several properties that list the players who have joined a session. The collection of all gamers in a session is described by the property [NetworkSession.AllGamers](#). This information is guaranteed to be the same on all game machines in a session. Within this list, the gamers might be local (playing on the same game machine through split-screen play) or remote (playing on a different game machine). The property [NetworkSession.LocalGamers](#) provides a list of all local players playing from a local game machine. The [NetworkSession.RemoteGamers](#) property provides a list of all players not playing on the local game machine.

To respond to players joining or leaving the game, you may subscribe to the [NetworkSession.GamerJoined](#) and [NetworkSession.GamerLeft](#) events with and create custom event handlers.

Subscribing to Gamer Events

To subscribe to gamer events

1. After creating or joining a session, subscribe to the events that are associated with players joining or leaving the game.

C#

```
session.GamerJoined +=
    new EventHandler<GamerJoinedEventArgs>(session_GamerJoined);
session.GamerLeft +=
    new EventHandler<GamerLeftEventArgs>(session_GamerLeft);
```

2. Create the [GamerJoined](#) event handler.

Note that the [GamerJoinedEventArgs](#) object contains a [Gamer](#) property. You can use it to read information about a specific player, or to associate information with a specific player. You can use the [Gamer.Tag](#) property to associate player data with the player.

C#

```
void session_GamerJoined(object sender, GamerJoinedEventArgs e)
{
    // Associate a tank object with this gamer.
    e.Gamer.Tag =
        new Tank(Content, GraphicsDevice.PresentationParameters);
}
```

3. Create the [GamerLeft](#) event handler.

C#

```
void session_GamerLeft(object sender, GamerLeftEventArgs e)
{
    Tank tank = e.Gamer.Tag as Tank;

    tank.LeaveGame();
}
```

Note

If the host player leaves the session calling [Game.Exit](#), this forces a host re-election. The host is automatically migrated to another player. If you are using a network topology other than peer-to-peer, such as a client-server or hybrid topology, and your host is also a server in the game, it will be necessary to migrate game state to the new game server when the host is migrated. Players in the game can use the [HostChanged](#) event to respond to host migration.

See Also

Concepts

[Network Session Management](#)

Tasks

[How To: Create a Network Session](#)

[How To: Find and Join a Network Session](#)

How To: Manage Player Movement Between Lobby and Gameplay Modes

Describes the properties and events associated with a multiplayer session that allow a game to move between lobby and gameplay modes. It identifies the game's current mode.

Managing Player Movement Between Modes

To manage player movement between lobby and gameplay modes

1. Hook up the session state change events.

The [NetworkSession.GameStarted](#) event indicates that all players have moved from the lobby to gameplay. The [NetworkSession.GameEnded](#) event occurs when the players move from gameplay back to the lobby. [NetworkSession.SessionEnded](#) occurs when the session is over, and all players move from the lobby of a session back to find another available session.

C#

```
session.GameStarted +=
    new EventHandler<GameStartedEventArgs>(session_GameStarted);
session.GameEnded +=
    new EventHandler<GameEndedEventArgs>(session_GameEnded);
session.SessionEnded +=
    new EventHandler<NetworkSessionEndedEventArgs>(
    session_SessionEnded);
```

2. Use [NetworkGamer.IsReady](#) to signal that a player is ready to move from the lobby to the game.

C#

```
// Signal I'm ready to play!
if (IsButtonPressed(GamePadButton.A))
{
    foreach (LocalNetworkGamer gamer in session.LocalGamers)
        gamer.IsReady = true;
}
```

3. On the host gaming machine, check [NetworkSession.IsEveryoneReady](#) to see if all players are ready.
4. If this is **true**, call [NetworkSession.StartGame](#) to change the game state from the lobby to gameplay mode.

C#

```
// The host checks if everyone is ready,
// and moves to game play if true.
if (session.IsHost)
{
    if (session.IsEveryoneReady)
    {
        session.StartGame();
        foreach (SignedInGamer signedInGamer in
            SignedInGamer.SignedInGamers)
        {
            signedInGamer.Presence.PresenceMode =
                GamerPresenceMode.InCombat;
        }
    }
}
```

5. In the [NetworkSession.GameStarted](#) event handler, make preparations for the game to begin.

In this example, the game object states are reset for the new game.

C#

```
void session_GameStarted(object sender, GameStartedEventArgs e)
{
    /// Reset everything when we are starting a new game.
    NetworkSession session = (NetworkSession)sender;

    for (int i = 0; i < session.AllGamers.Count; i++)
    {
        Tank tank = session.AllGamers[i].Tag as Tank;
        tank.Reset(i);
    }
}
```

6. In the [NetworkSession.GameEnded](#) event handler, make preparations to return the players to the lobby, such as recording the high score.

C#

```
void session_GameEnded(object sender, GameEndedEventArgs e)
{
    ReturnToLobby();
}
```

7. In the [NetworkSession.SessionEnded](#) event handler, return to the game title screen.

Note that before the session ends, the session can move between lobby and gameplay states as many times as the player wants. To retrieve the session state at any time, you can check the [NetworkSession.SessionState](#) property.

C#

```
void session_SessionEnded(object sender, NetworkSessionEndedEventArgs e)
{
    ReturnToTitleScreen();
}
```

See Also

Concepts

[Network Session Management](#)

Tasks

[How To: Create a Network Session](#)

[How To: Find and Join a Network Session](#)

[How To: Manage Players Joining and Leaving the Game](#)

How To: Join an In-Progress Game

Describes how to enable the [AllowJoinInProgress](#) property to make an in-progress game available to peers searching for available sessions.

Joining an In-Progress Game

To join an in-progress game

1. After creating a [NetworkSession](#), set [AllowJoinInProgress](#) to **true**.

C#

```
int maximumGamers = 8;
int privateGamerSlots = 2;
int maximumLocalPlayers = 1;

// Create the session
session = NetworkSession.Create(
    NetworkSessionType.SystemLink,
    maximumLocalPlayers, maximumGamers, privateGamerSlots,
    sessionProperties );

session.AllowJoinInProgress = true;
```

In the [NetworkSession.GamerJoined](#) event handler, the gamer may be a player who joined an in-progress game.

2. Associate any data necessary to enable a gamer to join while the game is in-progress.

C#

```
void session_GamerJoined(object sender, GamerJoinedEventArgs e)
{
    // Associate a tank object with this gamer.
    e.Gamer.Tag =
        new Tank(Content, GraphicsDevice.PresentationParameters);
}
```

Note

Network sessions of type [NetworkSessionType.Ranked](#) cannot be made join-in-progress, due to the competitive nature of ranked sessions.

See Also

Concepts

[Network Session Management](#)

Tasks

[How To: Manage Players Joining and Leaving the Game](#)

[How To: Find and Join a Network Session](#)

How To: Send Data

Describes the options available for sending data to all clients or to a specific player.

Sending Data

To send data to all peers

1. Create a [PacketWriter](#) to use in writing the data.

The [PacketWriter](#) is a helper for efficiently formatting outgoing network packets. A multiplayer game can create a single [PacketWriter](#) instance at startup, and reuse it any time the players want to send a packet.

C#

```
PacketWriter packetWriter = new PacketWriter();
```

Each [LocalNetworkGamer](#) playing on the same game machine might send data. At this point, players can loop through the [LocalGamers](#) collection.

2. To send a packet, call the various overloads of the [PacketWriter.Write](#) method to store data into the writer, and then pass the [PacketWriter](#) to [SendData](#).

Note that it is also possible to send data to a specific player by specifying the player in the call to [SendData](#).

Sending the packet will automatically clear the [PacketWriter](#). It can then be reused to write different data for another packet. Although it is not used in this example, the [PacketWriter](#) supports offsets through the [Position](#) property.

In the call to [SendData](#), be careful to specify a [SendDataOptions](#) value for *options* that is appropriate for the type of data being sent. Not all game data needs to be sent reliably. Sending excessive data using [SendDataOptions.ReliableInOrder](#) can cause the client to lag as it waits for data to be delivered in order.

C#

```
foreach (LocalNetworkGamer gamer in session.LocalGamers)
{
    // Get the tank associated with this player.
    Tank myTank = gamer.Tag as Tank;
    // Write the data.
    packetWriter.Write(myTank.Position);
    packetWriter.Write(myTank.TankRotation);
    packetWriter.Write(myTank.TurretRotation);
    packetWriter.Write(myTank.IsFiring);
    packetWriter.Write(myTank.Health);

    // Send it to everyone.
    gamer.SendData(packetWriter, SendDataOptions.None);
}
```

See Also

Tasks

[Third-Party Firewall Settings](#)

How To: Receive Data

Describes how the local players on a gaming machine receive data from the other players in a networked game.

Receiving Data

To receive data

1. Create a [PacketReader](#) to assist in reading the incoming network data.

A [PacketReader](#) is a helper for efficiently reading incoming network packets. A multiplayer game can create a single [PacketReader](#) instance at startup. It can reuse it any time the players want to read a packet.

C#

```
PacketReader packetReader = new PacketReader();
```

Each [LocalNetworkGamer](#) playing on the same game machine might receive data. At this point, players can loop through the [LocalGamers](#) collection.

2. To read a packet, pass the [PacketReader](#) to [ReceiveData](#), and then use the various [PacketReader.Read](#) methods to extract data from the reader.

C#

```
foreach (LocalNetworkGamer gamer in session.LocalGamers)
{
    // Keep reading while packets are available.
    while (gamer.IsDataAvailable)
    {
        NetworkGamer sender;

        // Read a single packet.
        gamer.ReceiveData(packetReader, out sender);

        if (!sender.IsLocal)
        {
            // Get the tank associated with this packet.
            Tank remoteTank = sender.Tag as Tank;

            // Read the data and apply it to the tank.
            remoteTank.Position = packetReader.ReadVector2();
            remoteTank.TankRotation = packetReader.ReadSingle();
            remoteTank.TurretRotation = packetReader.ReadSingle();
            remoteTank.IsFiring = packetReader.ReadBoolean();
            remoteTank.Health = packetReader.ReadInt32();
        }
    }
}
```

See Also

Tasks

[Third-Party Firewall Settings](#)

Hardware and Platforms

See the following sections for details about programming for the various hardware and platforms that are supported by the XNA Framework.

In This Section

[How To: Write Games for Less Capable Hardware](#)

Demonstrates how to write games for a variety of computer configurations. Specifically, this topic addresses ways to write a game so it will work on high-end hardware, yet continue to give adequate performance when the game is run on a computer that is less capable.

[Xbox 360 Programming](#)

Provides detailed information about the capabilities of the Xbox 360 hardware, and considerations that should be taken when programming games for the Xbox 360 platform. Includes a reference for Xbox 360–specific HLSL attributes and inline microcode.

[Zune Programming](#)

Provides information about Zune hardware capabilities and discusses Zune platform game programming.

[.NET Compact Framework for Xbox 360](#)

Provides information on developing XNA Game Studio applications using the C# language and the .NET Compact Framework for Xbox 360.

How To: Write Games for Less Capable Hardware

Demonstrates how to write games for a variety of computer configurations. Specifically, this topic addresses ways to write a game so it will work on high-end hardware, yet continue to give adequate performance when the game is run on a computer that is less capable.

Computers with different graphics cards and CPUs give different levels of performance. Writing game code that prepares for and adapts to slower and less capable hardware will allow games to have a larger audience.

Hardware Specific Degradation

Pixel Shader Support

When possible, take advantage of high-end hardware, but do not rely on it. For example, it is a good idea to create and use shaders to take advantage of higher-end hardware if it will benefit the game. But it is also a good idea to create alternative shaders that give acceptable effects and performance on lower-end hardware.

The following code sample tests the current graphics adapter. If the current graphics adapter does not support at least pixel shader version 2.0, a 1.1 version pixel shader is used instead.

C#

```
ContentManager content = new ContentManager(Services);
Effect effect;

// Check the graphics device used by the game for the
// necessary shader support.
GraphicsDeviceCapabilities caps =
    GraphicsDevice.GraphicsDeviceCapabilities;

if (caps.MaxPixelShaderProfile < ShaderProfile.PS_2_0)
{
    // Select a 1.1 version pixel shader if the computer has a
    // graphics adapter that cannot support 2.0 shaders.
    effect = content.Load<Effect>("PixelShader1_1");
}
else
{
    // Graphics adapter supports 2.0 version pixel shaders.
    effect = content.Load<Effect>("PixelShader2_0");
}
```

Single vs. Multiple CPUs

If a computer has multiple CPUs, some work can be offloaded to the other CPUs. The following code sample shows how to determine the number of available CPUs in the computer. If the game takes advantage of multiple CPUs, but does not require them, this code sample describes where to place code for single-processor computers and multi-processor computers.

C#

```
// Scale appropriately for the number of processors on this computer

if (Environment.ProcessorCount == 1)
{
    // Perform tasks specific to single processor systems.
}
else
{
    // Perform tasks specific to multi-processor systems.
}
```

Performance-Specific Degradation

You can measure performance levels by the number of times the [Update](#) method is called each second. Once the update rate falls below a certain level (the default is 60 times per second) [GameTime.IsRunningSlowly](#) will be set to **true**.

This code sample shows where to place code that should be skipped if the game's performance falls below 60 frames per second.

C#

```
// Perform the following only if the game is running
// at 60 frames per second or faster

if (gameTime.IsRunningSlowly == false)
{
    // Perform time consuming update or draw here.
}
```

If 60 frames per second is not ideal for the game, you can change the default by altering [TargetElapsedTime](#). To have [IsRunningSlowly](#) become **true** when the game slows to less than 30 frames per second, include this in the program's initialization code.

C#

```
// Alert the program when the frame rate falls below 30 frames per second.

TargetElapsedTime = new TimeSpan(10000000L / 30L);
```

To test for a different frame rate, change the 30 of the 30L to the frames-per-second desired (for example, 45L for 45 frames per second, or 100L for 100 frames per second).

See Also

Tasks

[How To: Check for Shader Model 2.0 Support](#)

Concepts

[Threading \(C# Programming Guide\)](#)

[Using Threading \(C# Programming Guide\)](#)

Reference

[IsRunningSlowly](#)

[TargetElapsedTime](#)

[System.Environment.ProcessorCount Property](#)

[ElapsedGameTime](#)

Xbox 360 Programming

The following documents provide detailed information about the capabilities of the Xbox 360 hardware, and list what to consider when programming games for the Xbox 360 platform.

In This Section

[Xbox 360 Programming Considerations](#)

Summarizes a few of the issues that XNA Framework programmers should consider when creating Xbox 360 games.

[Xbox 360 Device Capabilities](#)

This document enumerates the capabilities of the XNA Xbox 360 graphics device.

[Xbox 360 Surface Formats](#)

Lists the render target and depth-stencil surface formats that are available for the Xbox 360.

[Predicated Tiling](#)

Describes predicated tiling in Xbox 360 development. In predicated tiling, the commands issued in the **Draw** method are recorded before execution. The recorded commands, such as **DrawPrimitives** calls, are then executed for each tile, predicated based on whether the rendered primitives intersect the tile.

[HLSL Input Semantics \(Xbox 360\)](#)

Describes the high-level shader language (HLSL) input semantics available when developing games for Xbox 360.

[HLSL Attributes \(Xbox 360\)](#)

High-level shader language (HLSL) attributes are compiler hints that you can apply in your Xbox 360 shader code.

Xbox 360 Programming Considerations

Summarizes a few of the issues that XNA Framework programmers should consider when creating Xbox 360 games.

Graphics Considerations

The most important graphics consideration for programming for the Xbox 360 console is the wide variety of televisions that the Xbox 360 system supports. The Xbox 360 system supports both HDTV and normal television sets, at multiple resolutions (480p, 720p, 1080i, and 1080p), and multiple aspect ratios (4:3, 16:9, and 16:10). The console will automatically scale the output of a game to the resolution of the owner's display, and if the game is using a widescreen aspect ratio, the console automatically adds "black bars" (letterboxing) if the owner's display is not widescreen.

However, there are still two things XNA Framework programmers should consider when programming for the Xbox 360: the title safe region, and aspect ratios.

Title Safe Region

On standard tube televisions, the display area of the television is normally not a perfect rectangle. In fact, a significant amount of the display may not be visible on a CRT tube. In television, the inner 80–90 percent of the picture is considered the "title safe" region. Any graphics displayed outside of this region on a standard television may be obscured or distorted.

[How To: Draw a Sprite](#) demonstrates how to calculate the title safe region for the current display using the [Viewport](#) on the current [GraphicsDevice](#).

Programmers should ensure that any critical information (score, number of lives, ammo, and so on) is displayed within the title safe region, while drawing the background or a 3D scene across the entire display. Critical text should be displayed within the inner 80 percent of the screen. Programmers should also ensure that any text displayed on-screen is large enough to be legible on a standard television.

Aspect Ratio

If the aspect ratio of the back buffer is widescreen and the aspect ratio of the owner's display is not, the Xbox 360 will add black bars at the top of the display so that the entire back buffer is on screen. To avoid letterboxing, developers must program their games to use both standard and widescreen aspect ratios, and adjust their display to the default aspect ratio offered by the Xbox 360. For this reason, a single widescreen back buffer on Xbox games is recommended.

To detect the console display setting, you can use the [DisplayMode](#) property on the [GraphicsDevice](#). This will be valid during or after your game's [Initialize](#) method. Some common display modes are listed here.

Xbox Display Setting	DisplayMode Width and Height
AV (Composite)	640×480
480p (Normal)	640×480
480p (Widescreen)	640×480
720p (Widescreen)	1280×720
1080i/1080p (Widescreen)	1920×1080

Multisampling

4×AA multisampling is provided for free by the Xbox 360, so using 4×AA multisampling is recommended. Developers should be aware that this will often activate predicated tiling.

Shader Versions

Shaders compiled for Xbox 360 will be compiled as Shader 3.0 shaders regardless of the version specified in the effect file. Shaders written for 1.1 may exhibit errors on Xbox 360 if they depended on clamping behavior in 1.1 that is not present in 3.0.

Multiple Render Targets

Render targets function similarly on both Xbox 360 and Windows. By default, the content of a render target is always cleared when the render target is resolved unless [persistence](#) was specified during construction of the render target. Developers can avoid problems by rendering fully to each render target before using a new render target, and by rendering their final scene after all the other render targets have been resolved.

TextureUsage Flags

[TextureUsage](#) flag settings other than [TextureUsage.Tiled](#) and [TextureUsage.Linear](#) have no effect on Xbox 360. You will still get exceptions in Xbox 360 games if you specify [TextureUsage](#) settings that are not valid on Windows. It is recommended that you use [TextureUsage.None](#) on Xbox 360 and Windows whenever possible.

⚠ Caution

The [Texture.FromFile](#) method is only available when programming for Windows.

Dynamic Vertex Buffer Usage

Xbox 360 does not support dynamic vertex buffers that use the Overwrite/Discard semantic. This affects calls to [DrawPrimitives](#). For this reason, when rendering dynamic geometry, [DrawUserPrimitives](#) is recommended at all times on both Xbox 360 and Windows. (The performance disadvantages of [DrawUserPrimitives](#), though previously a factor on Windows, are largely obsolete in newer video drivers.)

Also for performance reasons, *static* geometry should always be rendered using vertex buffers and [DrawPrimitives](#).

For more information, see [Dynamic Vertex Buffers in XNA](#).

Input Considerations

The Xbox 360 supports three input devices: the game controller, the Xbox LIVE Vision camera, and an optional USB keyboard. Mice are not supported, and most Xbox 360 users will not have a keyboard connected to their console. The XNA Framework does not support the Vision camera. Games for the Xbox 360 should be programmed so that they will accept all critical input from the gamepad.

Audio Considerations

XACT audio wave banks, sound banks, and settings files are platform-specific. The XNA Framework Content Pipeline will automatically process XACT project files (.xap) into the correct platform type if the project file is added to Solution Explorer. If you are building the XACT project manually, ensure you are using the correct built files for your platform type when you load the built files into your game. Failure to use the correct platform-specific files will result in an error.

Storage Considerations

Accessing a player's storage on Xbox 360 (to let them save a game, for example) requires displaying the Storage Device Guide, where the user can choose between the hard drive or memory units. The best way to do this is to use the [BeginShowStorageDeviceSelector](#) method, as demonstrated in [How To: Get a StorageDevice Asynchronously](#). The largest space available on a memory unit is 52 MB, so any saved games must not exceed that size.

The maximum size of an XNA Framework project on an Xbox 360 console is 2 GB, so the entire game, including support files (level data, sound, and so on) must not exceed that size.

Source Code Considerations

Xbox 360 projects define an XBOX symbol for use with conditional compilation directives such as **#if**. You can use this to maintain source code that acts differently on Windows and Xbox 360 if that source is shared between projects. [How To: Draw a Sprite](#) shows an example of conditional compilation where the available screen space is calculated differently for Xbox 360 and Windows.

See Also

Concepts

[Developing Xbox 360 Games](#)

[Deploying an Xbox 360 Game](#)

[Debugging an Xbox 360 Game](#)

Xbox 360 Device Capabilities

This document enumerates the capabilities of the XNA Xbox 360 graphics device.

Xbox 360 Graphics Device Capabilities Property	Value	Note
AdapterOrdinalInGroup	0	Because the Xbox 360 has no multihead support, this value is disregarded.
AlphaCompareCapabilities		
SupportsNever	true	
SupportsLess	true	
SupportsEqual	true	
SupportsLessEqual	true	
SupportsGreater	true	
SupportsNotEqual	true	
SupportsGreaterEqual	true	
SupportsAlways	true	
CubeTextureFilterCapabilities		
SupportsMinifyPoint	true	
SupportsMinifyLinear	true	
SupportsMinifyAnisotropic	false	
SupportsMipMapPoint	true	
SupportsMipMapLinear	true	
SupportsMagnifyPoint	true	
SupportsMagnifyLinear	true	
SupportsMagnifyAnisotropic	false	
SupportsMagnifyPyramidalQuad	false	
SupportsMagnifyGaussianQuad	false	
SupportsMinifyPyramidalQuad	false	
SupportsMinifyGaussianQuad	false	
CursorCapabilities		
SupportsColor	false	
SupportsLowResolution	false	
DeclarationTypeCapabilities		
SupportsByte4	true	
SupportsRgba32	true	
SupportsNormalizedShort2	true	
SupportsNormalizedShort4	true	
SupportsRg32	true	
SupportsRgba64	true	
SupportsUInt101010	true	
SupportsNormalized101010	true	
SupportsHalfVector2	true	
SupportsHalfVector4	true	
DepthBufferCompareCapabilities		
SupportsNever	true	
SupportsLess	true	
SupportsEqual	true	
SupportsLessEqual	true	
SupportsGreater	true	
SupportsNotEqual	true	

SupportsGreaterEqual	true	
SupportsAlways	true	
DestinationBlendCapabilities		
SupportsZero	true	
SupportsOne	true	
SupportsSourceColor	true	
SupportsInverseSourceColor	true	
SupportsSourceAlpha	true	
SupportsInverseSourceAlpha	true	
SupportsDestinationAlpha	true	
SupportsInverseDestinationAlpha	true	
SupportsDestinationColor	true	
SupportsInverseDestinationColor	true	
SupportsSourceAlphaSat	false	
SupportsBothSourceAlpha	false	
SupportsBothInverseSourceAlpha	false	
SupportsBlendFactor	false	
DeviceCapabilities		
SupportsExecuteSystemMemory	false	
SupportsExecuteVideoMemory	false	
SupportsTransformedVertexSystemMemory	true	
SupportsTransformedVertexVideoMemory	true	
SupportsTextureSystemMemory	true	
SupportsTextureVideoMemory	true	
SupportsDrawPrimitivesTransformedVertex	true	
CanRenderAfterFlip	true	
SupportsTextureNonLocalVideoMemory	true	
SupportsDrawPrimitives2	false	
SupportsSeparateTextureMemories	false	
SupportsDrawPrimitives2Ex	false	
SupportsHardwareTransformAndLight	true	
CanDrawSystemToNonLocal	false	
SupportsHardwareRasterization	true	
IsDirect3D9Driver	true	
SupportsStreamOffset	true	
VertexElementScanSharesStreamOffset	false	
DeviceType	DeviceType.Hardware	
DriverCapabilities		
ReadScanLine	true	
SupportsFullScreenGamma	true	

CanCalibrateGamma	false	
CanManageResource	false	
SupportsDynamicTextures	true	
CanAutoGenerateMipMap	false	
SupportsAlphaFullScreenFlipOrDiscard	true	
SupportsLinearToSrgbPresentation	false	
SupportsCopyToVideoMemory	false	
SupportsCopyToSystemMemory	true	
ExtentsAdjust	0	
GuardBandBottom	0	
GuardBandLeft	0	
GuardBandRight	0	
GuardBandTop	0	
LineCapabilities		
SupportsTextureMapping	true	
SupportsDepthBufferTest	true	
SupportsBlend	true	
SupportsAlphaCompare	true	
SupportsFog	false	
SupportsAntiAlias	false	
MasterAdapterOrdinal	0	Because the Xbox 360 has no multihead support, this value is disregarded.
MaxAnisotropy	16	
MaxPixelShader30InstructionSlots	4096	
MaxPixelShaderProfile	XPS_3_0	
MaxPointSize	256	
MaxPrimitiveCount	1048575	
MaxSimultaneousRenderTarget	4	
MaxSimultaneousTextures	16	
MaxStreams	16	
MaxStreamStride	1024	
MaxTextureAspectRatio	8192	
MaxTextureHeight	8192	This value reflects 2D textures only.
MaxTextureRepeat	2048	
MaxTextureWidth	8192	This value reflects 2D textures only.
MaxUserClipPlanes	6	
MaxVertexIndex	16777215	
MaxVertexShader30InstructionSlots	4096	
MaxVertexShaderConstants	256	
MaxVertexShaderProfile	XVS_3_0	
MaxVertexW	1E+10	
MaxVolumeExtent	1024	
NumberOfAdaptersInGroup	1	Because the Xbox 360 has no multihead support, this value is disregarded.
PixelShader1xMaxValue	3.402823E+38	
PixelShaderCapabilities		
SupportsPredication	true	

SupportsArbitrarySwizzle	true	
SupportsGradientInstructions	true	
SupportsNoDependentReadLimit	true	
SupportsNoTextureInstructionLimit	true	
DynamicFlowControlDepth	4	
NumberTemps	64	
StaticFlowControlDepth	4	
NumberInstructionSlots	2048	
PixelShaderVersion	3.0	
PresentInterval	PresentInterval.One PresentInterval.Two PresentInterval.Three PresentInterval.Four PresentInterval.Immediate	
PrimitiveCapabilities		
SupportsMaskZ	true	
SupportsCullNone	true	
SupportsCullClockwiseFace	true	
SupportsCullCounterClockwiseFace	true	
SupportsColorWrite	true	
SupportsClipTransformedVertices	false	
SupportsBlendOperation	true	
IsNullReference	false	
SupportsIndependentWriteMasks	false	
SupportsFogAndSpecularAlpha	false	
SupportsSeparateAlphaBlend	true	
SupportsMultipleRenderTargetIndependentBitDepths	false	
SupportsMultipleRenderTargetPostPixelShaderBlending	false	
HasFogVertexClamped	false	
RasterCapabilities		
SupportsDepthBufferTest	true	
SupportsFogVertex	false	
SupportsFogTable	false	
SupportsMipMapLevelOfDetailBias	true	
SupportsDepthBufferLessHsr	true	
SupportsFogRange	false	
SupportsAnisotropy	true	
SupportsWFog	false	
SupportsDepthFog	false	
SupportsColorPerspective	true	
SupportsScissorTest	true	
SupportsSlopeScaleDepthBias	false	
SupportsDepthBias	false	
SupportsMultisampleToggle	true	
ShadingCapabilities		

SupportsColorGouraudRgb	true	
SupportsSpecularGouraudR gb	true	
SupportsAlphaGouraudBlen d	true	
SupportsFogGouraud	false	
SourceBlendCapabilities		
SupportsZero	true	
SupportsOne	true	
SupportsSourceColor	true	
SupportsInverseSourceColor	true	
SupportsSourceAlpha	true	
SupportsInverseSourceAlph a	true	
SupportsDestinationAlpha	true	
SupportsInverseDestination Alpha	true	
SupportsDestinationColor	true	
SupportsInverseDestination Color	true	
SupportsSourceAlphaSat	false	
SupportsBothSourceAlpha	false	
SupportsBothInverseSource Alpha	false	
SupportsBlendFactor	false	
StencilCapabilities		
SupportsKeep	true	
SupportsZero	true	
SupportsReplace	true	
SupportsIncrementSaturatio n	true	
SupportsDecrementSaturati on	true	
SupportsInvert	true	
SupportsIncrement	true	
SupportsDecrement	true	
SupportsTwoSided	true	
TextureAddressCapabilities		
SupportsWrap	true	
SupportsMirror	true	
SupportsClamp	true	
SupportsBorder	false	
SupportsIndependentUV	true	
SupportsMirrorOnce	true	
TextureCapabilities		
SupportsPerspective	true	
SupportsAlpha	true	
RequiresPower2	false	
RequiresSquareOnly	false	
SupportsTextureRepeatNotS caledBySize	true	
SupportsNonPower2Conditio nal	false	
SupportsProjected	true	
SupportsCubeMap	true	

SupportsVolumeMap	true	
SupportsMipMap	true	
SupportsMipVolumeMap	true	
SupportsMipCubeMap	true	
RequiresCubeMapPower2	false	
RequiresVolumeMapPower2	false	
SupportsNoProjectedBumpEnvironment	false	
TextureFilterCapabilities		
SupportsMinifyPoint	true	
SupportsMinifyLinear	true	
SupportsMinifyAnisotropic	true	
SupportsMipMapPoint	true	
SupportsMipMapLinear	true	
SupportsMagnifyPoint	true	
SupportsMagnifyLinear	true	
SupportsMagnifyAnisotropic	true	
SupportsMagnifyPyramidalQuad	false	
SupportsMagnifyGaussianQuad	false	
SupportsMinifyPyramidalQuad	false	
SupportsMinifyGaussianQuad	false	
VertexFormatCapabilities		
NumberSimultaneousTextureCoordinates	8	
SupportsDoNotStripElements	false	
SupportsPointSize	true	
VertexProcessingCapabilities		
SupportsTextureGeneration	true	
SupportsLocalViewer	true	
SupportsTextureGenerationSphereMap	true	
SupportsNoTextureGenerationNonLocalViewer	false	
VertexShaderCapabilities		
SupportsPredication	true	
DynamicFlowControlDepth	4	
NumberTemps	64	
StaticFlowControlDepth	4	
VertexShaderVersion	3.0	
VertexTextureFilterCapabilities		
SupportsMinifyPoint	true	
SupportsMinifyLinear	true	
SupportsMinifyAnisotropic	true	
SupportsMipMapPoint	true	
SupportsMipMapLinear	true	
SupportsMagnifyPoint	true	
SupportsMagnifyLinear	true	
SupportsMagnifyAnisotropic	true	
SupportsMagnifyPyramidalQuad	false	

SupportsMagnifyGaussianQuad	false	
SupportsMinifyPyramidalQuad	false	
SupportsMinifyGaussianQuad	false	
VolumeTextureAddressCapabilities		
SupportsWrap	true	
SupportsMirror	true	
SupportsClamp	true	
SupportsBorder	false	
SupportsIndependentUV	true	
SupportsMirrorOnce	true	
VolumeTextureFilterCapabilities		
SupportsMinifyPoint	true	
SupportsMinifyLinear	true	
SupportsMinifyAnisotropic	false	
SupportsMipMapPoint	true	
SupportsMipMapLinear	true	
SupportsMagnifyPoint	true	
SupportsMagnifyLinear	true	
SupportsMagnifyAnisotropic	false	
SupportsMagnifyPyramidalQuad	false	
SupportsMagnifyGaussianQuad	false	
SupportsMinifyPyramidalQuad	false	
SupportsMinifyGaussianQuad	false	

See Also [Xbox 360 Programming Considerations](#)

Xbox 360 Surface Formats

In the XNA Framework, all two-dimensional (2D) images are represented by a range of memory called a surface. Within a surface, each element holds a color value representing a small section of the image, called a pixel. An image's detail level is defined by both the number of pixels needed to represent the image, and the number of bits needed for the image's color spectrum. For example, an image that is 800 pixels wide by 600 pixels high with 32 bits of color for each pixel (written as $800 \times 600 \times 32$) is more detailed than an image that is 640 pixels wide by 480 pixels tall with 16 bits of color for each pixel (written as $640 \times 480 \times 16$). Likewise, the more detailed image requires a larger surface to store the data. For an $800 \times 600 \times 32$ image, the surface's array dimensions are 800×600 , and each element holds a 32-bit value to represent its color.

All surfaces have a size and store a specific number of bits that represent color. The bits that represent color are separated into individual color elements: red, green, and blue. All color elements are defined by the [SurfaceFormat](#) enumerated type. A color format is broken down into the number of bytes reserved for each color. For example, a 16-bit color format is defined as [SurfaceFormat.BGR565](#), where 5 bits are reserved for red (R), 6 bits for green (G), and 5 bits for blue (B).

On any platform, you can obtain the available render target platforms and depth-stencil formats by using the [GraphicsAdapter.CheckDeviceFormat](#) and [GraphicsAdapter.CheckDepthStencilMatch](#) methods

Render Target Formats

The following render target formats are supported on the Xbox 360 platform.

- [SurfaceFormat.Single](#)
- [SurfaceFormat.Vector2](#)
- [SurfaceFormat.HalfVector2](#)
- [SurfaceFormat.HalfVector4](#)
- [SurfaceFormat.Color](#)
- [SurfaceFormat.Bgr32](#)
- [SurfaceFormat.Rgba32](#)
- [SurfaceFormat.Rgb32](#)
- [SurfaceFormat.Bgra1010102](#)
- [SurfaceFormat.Rgba1010102](#)
- [SurfaceFormat.NormalizedAlpha1010102](#)

Both $2 \times$ and $4 \times$ multisampling are supported. On resolve, the GPU can downsample multisampled render targets, and it can format conversion. However, multisample downsampling is limited to blendable target formats.

Render targets must be padded to certain dimensions based on the multisample mode. The padding required for 32-bits-per-pixel formats is 80×16 for $1 \times$ antialiasing, 80×8 for $2 \times$ antialiasing, and 40×8 for $4 \times$ antialiasing.

Depth-Stencil Formats

The following depth-stencil formats are supported on the Xbox 360 platform.

- [DepthFormat.Depth24](#)
- [DepthFormat.Depth24Stencil8](#)
- [DepthFormat.Depth24Stencil8Single](#)

Unsupported Surface Formats

The following surface formats are not supported on the Xbox 360 platform for any [ResourceType](#), including textures:

- [SurfaceFormat.Bgr24](#)
- [SurfaceFormat.Bgra2338](#)
- [SurfaceFormat.Bgr233](#)
- [SurfaceFormat.NormalizedByte2Computed](#)
- [SurfaceFormat.LuminanceAlpha8](#)
- [SurfaceFormat.Palette8](#)
- [SurfaceFormat.PaletteAlpha16](#)
- [SurfaceFormat.Multi2Bgra32](#)

- [SurfaceFormat.Depth15Stencil1](#)
- [SurfaceFormat.Depth24Stencil4](#)

See Also

Reference

[GraphicsAdapter.CheckDepthStencilMatch Method](#)

[GraphicsAdapter.CheckDeviceFormat Method](#)

[SurfaceFormat Enumeration](#)

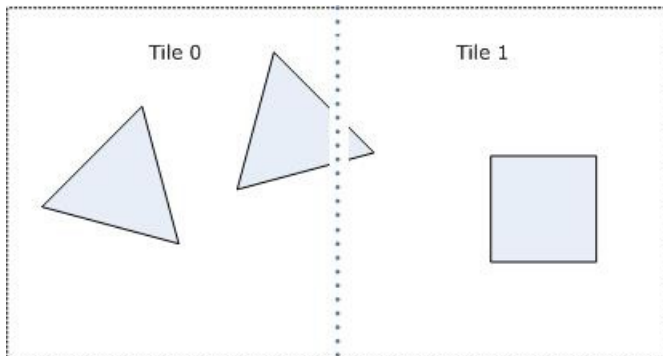
[DepthFormat Enumeration](#)

Predicated Tiling

Describes predicated tiling in Xbox 360 development.

The Xbox 360 has 10 MB (10×1024×1024) of fast embedded dynamic RAM (EDRAM) that is dedicated for use as the back buffer, depth stencil buffer, and other render targets. Depending on the size and format of the render targets and the antialiasing level, it may not be possible to fit all targets in EDRAM at once. For example, 10 MB of EDRAM is enough to hold two 1280×720 32-bit surfaces with no multisample antialiasing (MSAA) or two 640×480 4× MSAA 32-bit surfaces. However, a 1280×720 2× MSAA 32-bits-per-pixel render target is 7,372,800 bytes. Combined with a 32-bit Z/stencil buffer of the same dimensions, it becomes apparent that 10 MB might not be sufficient.

Predicated tiling allows rendering to larger surfaces than can fit into EDRAM at any one time. In predicated tiling, the screen space is broken up into tiles (rectangles). The following figure shows the screen space broken into two tiles.



In predicated tiling, the commands issued in the **Draw** method are recorded before execution. The recorded commands, such as **DrawPrimitives** calls, are then executed for each tile, predicated based on whether the rendered primitives intersect the tile. In the preceding figure, both the triangle primitives would be rendered in **Tile 0**. Once the primitives for a tile are fully rendered, the tile is then resolved into the texture that is used for the front buffer. Each successive tile is handled the same way and is resolved into the same texture.

Triggering Predicated Tiling

Predicated tiling occurs automatically for Xbox 360 games created with XNA Game Studio when the size and format of the render targets exceed the console's available EDRAM. In predicated tiling, the size of the render targets and the depth-stencil are no longer limited by EDRAM memory, although each individual tile has to fit into EDRAM.

Once predicated tiling has been triggered, all rendering commands in **Draw** are accumulated and played back for each tile in separate passes. An adjusted window offset and clip rectangle are used to render only those portions that intersect with the specified screen-space tile rectangle. Drawing is done only when the primitive drawing call is known to be visible on the given tile. This is determined by using hardware screen-space extent queries, which compute if the geometry lies within the tile. All rendering is then completed using the render target and depth stencil surface that are currently set.

Restrictions Encountered After Predicated Tiling Has Been Triggered

A few restrictions apply when predicated tiling has been triggered.

Resources that are referenced in the command buffer cannot be modified by the CPU, because the command buffer has to be played back more than once to accomplish the tiling. It is okay for the GPU to modify resources and then reuse them in the same pass, because the GPU can appropriately recreate the memory on every pass.

Specifically, in the **Draw** method of an Xbox 360 game, vertex buffers, index buffers, and textures should not be written to using **SetData** during the **Draw** method. This condition may lead to graphics corruption or crashes. For vertex buffers, this issue can be avoided by using [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#) as the preferred alternative to [VertexBuffer.SetData](#) for dynamic vertex generation.

Data from an occlusion query (specifically, a call to [IsComplete](#)) on Xbox 360 is not available from within the tiling bracket. Access this data by switching to a different render target or end the frame. This condition does not apply to games targeting the Windows platform.

HLSL Input Semantics (Xbox 360)

Describes the high-level shader language (HLSL) input semantics available when developing games for Xbox 360.

Vertex Shader Input Semantics

One special vertex shader input semantic is available in HLSL for the Xbox 360: INDEX. Any kind of shader can use the INDEX input semantic. It enables you to retrieve vertex.

For more details, see [Vertex Fetching in HLSL](#).

Pixel Shader Input Semantics

To support point sprites, a specific pixel shader input semantic is available in HLSL: SPRITETEXCOORD. This semantic is required to read the sprite texture coordinates in a point sprite pixel shader.

The following HLSL code demonstrates a pixel shader that can be used with SPRITETEXCOORD to render point sprites.

```
sampler2D s;  
float4 main( float2 t : SPRITETEXCOORD ) : COLOR  
{  
    return tex2D( s, t );  
}
```

See Also

Tasks

[How To: Draw Point Sprites](#)

Concepts

[Xbox 360 Programming Considerations](#)

[HLSL Attributes \(Xbox 360\)](#)

[Microcode \(xvs_3_0, xps_3_0\)](#)

[Shader Content Catalog at XNA Creators Club Online](#)

HLSL Attributes (Xbox 360)

High-level shader language (HLSL) attributes are compiler hints that you can apply in your Xbox 360 shader code.

The HLSL compiler uses attributes to generate microcode in the manner specified by the attributes. HLSL attributes give the compiler more information about how to compile your shaders.

Use HLSL attributes to generate optimized microcode from HLSL source. For example, you can use HLSL attributes to control how branching statements are compiled into microcode, controlling the type microcode branching instructions that will be emitted. Using HLSL attributes, you can optimize your code and sometimes control the amount of microcode emitted.

HLSL attributes for Xbox 360 are similar to DirectX 10 HLSL attributes for Microsoft Windows.

In This Section

[Attribute Categories](#)

High-level shader language (HLSL) attributes for Xbox 360 are compiler hints that influence how microcode is generated from HLSL. Using the attributes, you can optimize how microcode is emitted and tune the shader performance.

[Attribute Syntax](#)

High level shader language (HLSL) attributes use the following syntax.

[HLSL Attributes Reference \(Xbox 360\)](#)

Contains reference information about high-level shader language (HLSL) attributes. The following HLSL attributes are implemented.

See Also

Concepts

[Shader Content Catalog at XNA Creators Club Online](#)

Attribute Categories

High-level shader language (HLSL) attributes for Xbox 360 are compiler hints that influence how microcode is generated from HLSL. Using the attributes, you can optimize how microcode is emitted and tune the shader performance.

The high-level shader language attributes consist of the following categories.

- [Branching Statement Attributes](#)
- [Looping Statement Attributes](#)
- [Function Attributes](#)
- [Miscellaneous Attributes](#)

Branching Statement Attributes

HLSL supports the following attributes that modify the microcode output of branching statements. These attributes affect only **if** statements.

- [branch](#)
- [flatten](#)
- [ifAll](#)
- [ifAny](#)
- [predicate](#)
- [predicateBlock](#)

Looping Statement Attributes

HLSL supports the following attributes that modify the microcode output of looping statements. Use these attributes to affect **for**, **while**, and **do-while** statements.

- [unroll](#)
- [loop](#)

Function Attributes

HLSL supports the following attributes that modify the microcode output related to functions. Use these attributes to affect either how functions are called or how microcode is generated throughout the scope of a function.

- [call](#)
- [maxExports](#)
- [maxTempReg](#)
- [reduceTempRegUsage](#)
- [smpreg](#)

Miscellaneous Attributes

HLSL supports the following attributes to isolate areas of HLSL code for optimization and mark certain input parameters as unused.

- [isolate](#)
- [maxInstructionCount](#)
- [noExpressionOptimizations](#)
- [removeUnusedInputs](#)
- [unused](#)
- [xps](#)

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

Attribute Syntax

High level shader language (HLSL) attributes use the following syntax.

```
[AttributeName [ (Parameter) ] ]
```

Apply HLSL attributes by specifying the attribute name between opening and closing brackets.

Some attributes accept parameters, while others do not. Parameters are passed between opening and closing parentheses similar to the way C function parameters are passed. If the attribute does not accept a parameter, you do not have to supply the parentheses.

Different HLSL attributes modify different statement types. To modify a statement, place the attribute in front of a statement type that is compatible with the particular attribute you are using. HLSL attributes can modify the following types of statements.

- Branching statements.
- Loop statements.
- Functions.
- Function parameters.
- Output parameters.
- Arbitrary statements, functions, or scopes delineated by brackets.
- Entire shaders.

Applying HLSL Attributes

The following code example demonstrates how to apply an attribute. This example applies the `call` attribute to an HLSL function. Using the `call` attribute, you can specify that a function should not be inlined.

```
[call]
float4 MyFunction()
{
// ...
}
```

The HLSL compiler provides warnings that help you properly apply attributes. The HLSL compiler displays a warning in the following cases.

- An attribute is unrecognized.
- An attribute is applied to an incompatible statement.
- An attribute cannot be respected in the current context.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

HLSL Attributes Reference (Xbox 360)

Contains reference information about high-level shader language (HLSL) attributes. The following HLSL attributes are implemented.

Attribute	Description
branch	Performs branching.
call	Prevents inlining of a function.
flatten	Performs branching.
ifAll	Executes the conditional part of an if statement when the condition is true for all threads on which the current shader is running.
ifAny	Executes the conditional part of an if statement when the condition is true for any thread on which the current shader is running.
isolate	Optimizes the specified HLSL code independently of the surrounding code.
loop	Gives preference to flow control constructs.
maxexports	Specifies the maximum number of export instructions that will execute along any path from the entry point to an exit.
maxInstructionCount	Sets the maximum number of instructions available to a shader.
maxtempreg	Restricts temporary register usage to the number of registers specified. Generates a compiler error if unsuccessful.
noExpressionOptimizations	Avoids optimization of expressions.
predicate	Performs branching by using predication.
predicateBlock	Performs branching by using predicated exec blocks.
reduceTempRegUsage	Restricts temporary register usage to the number of registers specified. Generates a compiler warning if unsuccessful.
removeUnusedInputs	Removes unused interpolator inputs from pixel shaders.
sampreg	Sets the ranges of pixel and vertex sampler registers used by the compiler.
unroll	Avoids flow control constructs.
unused	Suppresses warnings about unused shader parameters.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

branch (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Performs branching using control flow instructions.

Syntax

```
[branch]
```

Parameters

None

Scope

Applies to **if** statements.

Examples

The following HLSL code snippet demonstrates how to apply the **branch** attribute.

```
[branch]
if( a )
{
    a = sqrt( a );
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

call (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Prevents inlining of a function.

Syntax

```
[call]
```

Parameters

None

Scope

Applies to function definitions and function calls.

Remarks

By default, all user-defined HLSL functions are inlined by the HLSL compiler on Xbox 360. Use the **call** attribute to turn off inlining and decrease the amount of microcode emitted by the HLSL compiler.

You can use the **call** attribute to control inlining either when the function is defined or when a function is called.

- If applied to a function definition, the function will not be inlined anywhere.
- If applied to a function call, the function will not be inlined during the current call, but might be inlined elsewhere.

Examples

The following HLSL code snippet demonstrates how to apply the **call** attribute.

```
// This function will not be inlined.
[call]
float4 ComputePosition1 ()
{
    ...
}

// This function will be inlined wherever possible.
float4 ComputePosition2 ()
{
    ...
}

float4 main () : POSITION
{
    // ComputePosition1 will not be inlined.
    // ComputePosition2 will be inlined at this specific call site.
    float4 pos = ComputePosition1() + ComputePosition2();

    // ComputePosition2 will not be inlined at this specific call site.
    [call]
    return pos + ComputePosition2();
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

flatten (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Performs branching using conditional move instructions.

Syntax

```
[flatten]
```

Parameters

None.

Scope

Applies to **if** statements.

Examples

The following HLSL code snippet shows how to apply the **flatten** attribute.

```
[flatten]
if( a )
{
    a = sqrt( a );
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

ifAll (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Executes the conditional part of an **if** statement when the condition is true for all threads on which the current shader is running.

Syntax

```
[ifAll]
```

Parameters

None.

Scope

Applies to **if** statements.

Remarks

Apply the **ifAll** attribute to **if** statements to make a dynamic branch behave more like a static branch. This causes all threads to branch together. The **ifAll** attribute executes conditional statements only if the condition is true on all threads.

On static branches, the **ifAll** attribute is similar to the [branch](#) attribute. However, the **ifAll** attribute does not permit optimizations that need to invert the original branch condition.

On dynamic branches, the **ifAll** attribute differs from the [branch](#) attribute. When you apply the [branch](#) attribute to a dynamic **if** statement, the conditional statements are predicated. They are skipped only if no thread needs to execute the code. The [branch](#) attribute allows each thread to execute different instructions for an **if** statement. The **ifAll** attribute forces every thread to execute the same instructions.

Note that you cannot apply the **ifAll** attribute to a nested dynamic branch.

Examples

The following HLSL code snippet shows how to apply the **ifAll** attribute.

```
float x = tex2D( sampler, coord );
float result = g_value1;

[ifAll]
if (x > 0)
{
    result = g_value2;
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[ifAny \(HLSL\)](#)

[branch \(HLSL\)](#)

[HLSL Attributes Reference \(Xbox 360\)](#)

ifAny (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Executes the conditional part of an **if** statement when the condition is true for any thread on which the current shader is running.

Syntax

```
[ifAny]
```

Parameters

None.

Scope

Applies to **if** statements.

Remarks

Apply the **ifAny** attribute to **if** statements to make a dynamic branch behave more like a static branch. This causes all threads to branch together. The **ifAny** attribute executes conditional statements if the condition is true on any thread.

On static branches, the **ifAny** attribute is similar to the [branch](#) attribute. However, the **ifAny** attribute does not permit optimizations that need to invert the original branch condition.

On dynamic branches, the **ifAny** attribute differs from the [branch](#) attribute. When you apply the [branch](#) attribute to a dynamic **if** statement, the conditional statements are predicated. They are skipped only if no thread needs to execute the code. The [branch](#) attribute allows each thread to execute different instructions for an **if** statement. The **ifAny** attribute forces every thread to execute the same instructions.

Note that you cannot apply the **ifAny** attribute to a nested dynamic branch.

Examples

The following HLSL code snippet shows how to apply the **ifAny** attribute.

```
float x = tex2D( sampler, coord );
float result = g_value1;
[ifAny]
if (x > 0)
{
    result = g_value2;
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[ifAll \(HLSL\)](#)

[branch \(HLSL\)](#)

[HLSL Attributes Reference \(Xbox 360\)](#)

isolate (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Optimizes the specified HLSL code independently of the surrounding code.

Syntax

```
[isolate]
```

Parameters

None.

Scope

Applies to output parameter declarations, functions, structures, or any scope.

Remarks

You can apply the **isolate** attribute in the following ways.

- When the attribute is applied to an output parameter declaration, any code related to computation of the output is optimized independently of the rest of the function.
- When the attribute is applied to a scope, microcode in the scope is optimized independently of the surrounding code. Isolated scopes are ordered with respect to each other. Isolated scopes can be nested. The inner scope is optimized independently of the outer scope.
- When the attribute is applied to a function, the function can be inlined, but the inlined code will not be merged with code at the call site.
- When the attribute is applied to a structure that is returned by a shader, the function has the same effect as applying it to an output parameter.

The **isolate** attribute is useful for multipass shaders that need a floating-point output of two different shaders to be equivalent bitwise. The generated microcode of the two shaders might not be identical bit-for-bit, but both shaders should compute bit-for-bit identical results.

When you work with multipass shaders, the source code affected by the **isolate** attribute must be identical to guarantee identical microcode. Subtle differences in the HLSL can cause the **isolate** attribute to emit different microcode. Sometimes the emitted microcode has the same instructions, but it might have different [swizzle](#) values, or it might have different combinations of the same vector computations. Differences in HLSL can also affect the order in which optimizations occur.

Example

The following HLSL code snippet shows how to apply the **isolate** attribute.

```
// Apply the attribute to a declaration.
void main ([isolate] out float4 pos : POSITION)
{
    pos = ComputePosition();
}

// These two fetches are always generated in the order specified in the
// HLSL.
[isolate]
{
    tex1 = tex3D( coord1 );
}

[isolate]
{
    tex2 = tex3D( coord2 );
}

// The HLSL compiler can swap these two fetches, but the compiler cannot
// insert other instructions between them.
```

```
[isolate]
{
    tex1 = tex3D( coord1 );
    tex2 = tex3D( coord2 );
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

loop (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Gives preference to flow control constructs.

Syntax

```
[loop]
```

Parameters

None.

Scope

Applies to **for**, **while**, and **do-while** statements.

Examples

The following HLSL code snippet shows how to apply the **loop** attribute.

```
float value = 0;

[loop]
for( uint i = 0; i < 4; i++ )
{
    value ++;
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

maxexports (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Specifies the maximum number of export instructions that will execute along any path from the entry point to an exit.

Syntax

```
[maxexports(n)]
```

Parameters

n

Integer that represents the maximum number of export instructions.

Scope

Applies to functions. The **maxexports** attribute is ignored on any function except the shader entry point that is currently being compiled.

Examples

The following HLSL code snippet shows how to apply the **maxexports** attribute.

```
row_major float4x4 a;  
float4 b;  
row_major float4x4 c;  
float4 d;  
  
[maxexports(2)]  
void main ( out float4 pos : POSITION,  
            out float4 coord : TEXCOORD ) {  
  
    pos = mul( a, b );  
    coord = mul( c, d );  
}
```

Without the call to the **maxexports** attribute, each **mul** HLSL intrinsic yields four microcode instructions.

With the call to the **maxexports** attribute, the HLSL compiler adds extra **mov** instructions to reduce the number of export instructions to two.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

maxInstructionCount (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Sets the maximum number of instructions available to a shader.

Syntax

```
[maxInstructionCount(n)]
```

Parameters

n

Integer that represents the maximum number of instructions. The *n* parameter must be a value between 256 and 3839.

Scope

Applies to an entire shader.

Remarks

The Xbox 360 GPU contains an instruction store of 4096 microcode instructions that Direct3D allocates between vertex and pixel shaders. By default, Direct3D allocates 2047 instructions on the GPU to vertex shaders and 2048 instructions on the GPU to pixel shaders. (One instruction at index 0 is reserved.) By default, the HLSL compiler accommodates these requirements by emitting shaders that do not exceed 2048 instructions.

Use the **maxInstructionCount** attribute to increase or decrease the limit to the number of instructions that can be emitted by the HLSL compiler. Note that this attribute does not change the number of instructions allocated to shaders on the GPU.

The number of instructions that can be emitted by the HLSL compiler should not be greater than the number of instructions available on the GPU. This situation produces an error.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

maxtempreg (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Restricts temporary register usage to the number of registers specified. Generates a compiler error if unsuccessful.

Syntax

```
[maxtempreg(n)]
```

Parameters

n

Integer that represents the maximum number of temporary registers.

Scope

Applies to functions. The **maxtempreg** attribute is ignored on any function except the shader entry point that is currently being compiled. The parameter also applies to any function that is compiled for this entry point, whether the function is inlined or not.

Remarks

If the HLSL compiler is unable to restrict temporary registers to the number specified by the **maxtempreg** attribute, the compiler reports an error and fails to compile the shader.

Note that the [reduceTempRegUsage](#) attribute also restricts temporary register usage. However, the [reduceTempRegUsage](#) attribute reports a warning and then does compile the shader by using the number of temporary registers as close as possible to the number specified.

Examples

The following HLSL code snippet shows how to apply the **maxtempreg** attribute and the [reduceTempRegUsage](#) attribute. This example specifies that the compiler never allocate more than 32 registers for temporary usage. Instead, the compiler should try to stay as close to 12 registers as possible. If the compiler exceeds an allocation of 12 registers, it displays a warning. However, if the compiler exceeds 32 registers, it displays an error and fails to compile the shader.

```
[maxtempreg (32) ]  
[reduceTempRegUsage (12) ]  
float4 main ( float4 in )  
{  
    // ...  
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[reduceTempRegUsage](#)

[HLSL Attributes Reference \(Xbox 360\)](#)

noExpressionOptimizations (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Avoids optimization of expressions.

Syntax

```
[noExpressionOptimizations]
```

Parameters

None.

Scope

Applies to statements, functions, or any scope.

Remarks

The **noExpressionOptimizations** attribute instructs the compiler not to optimize expressions. The resulting microcode assembly is kept as close to the original HLSL as possible.

In general, do not apply the **noExpressionOptimizations** attribute. Use it only to disable optimizations that are causing problems in your code.

In some situations, the compiler must perform expression optimization even if you apply the **noExpressionOptimizations** attribute. For example, the compiler must optimize expressions during sampler array indexing.

Also, the compiler may need to optimize expressions within the scope of the attribute when neighboring code is optimized. To avoid this situation, apply the **noExpressionOptimizations** attribute to the shader entry point.

The **noExpressionOptimizations** attribute cannot be nested. This attribute implies the [isolate](#) attribute.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

predicate (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Performs branching by using predication.

Syntax

```
[predicate]
```

Parameters

None.

Scope

Applies to **if** statements.

Remarks

Instructions are individually predicated.

Examples

The following HLSL code snippet shows how to apply the **predicate** attribute.

```
[predicate]
if( a )
{
    a = sqrt( a );
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

predicateBlock (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Performs branching by using predicated **exec** blocks.

Syntax

```
[predicateBlock]
```

Parameters

None.

Scope

Applies to **if** statements.

Remarks

Use the **predicateBlock** attribute with the [flatten](#) attribute to specify exactly how to flatten a branch.

Examples

The following HLSL code snippet shows how to apply the **predicateBlock** attribute.

```
[predicateBlock]
if( a )
{
    a = sqrt( a );
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

reduceTempRegUsage (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Restricts temporary register usage to the number of registers specified. Generates a compiler warning if unsuccessful.

Syntax

```
[reduceTempRegUsage(n)]
```

Parameters

n

Integer that represents the maximum number of temporary registers.

Scope

Applies to functions. The **reduceTempRegUsage** attribute is ignored on any function except the shader entry point that is currently being compiled.

Remarks

If the HLSL compiler is unable to restrict temporary registers to the number specified by the **reduceTempRegUsage** attribute, the HLSL compiler reports a warning and then compiles by using the number of temporary registers as close as possible to the number specified.

Note that the [maxtempreg](#) attribute also restricts temporary register usage. However, the [maxtempreg](#) attribute reports an error and fails to compile if it is unable to restrict temporary registers to the number specified.

Examples

The following HLSL code snippet demonstrates how to apply the **reduceTempRegUsage** attribute and the [maxtempreg](#) attribute. This example specifies that the compiler should never allocate more than 32 registers for temporary usage, but should try to stay as close to 12 registers as possible. If the compiler exceeds an allocation of 12 registers, it displays a warning. However, if the compiler exceeds 32 registers, it displays an error and fails to compile the shader.

```
[maxtempreg(32) ]
[reduceTempRegUsage(12) ]
float4 main ( float4 in )
{
    // ...
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[maxtempreg](#)

[HLSL Attributes Reference \(Xbox 360\)](#)

removeUnusedInputs (HLSL)

Removes unused interpolator inputs from pixel shaders.

Syntax

```
[removeUnusedInputs]
```

Parameters

None.

Scope

Applies to an entire pixel shader.

Remarks

Apply the **removeUnusedInputs** attribute to a pixel shader entry point. This automatically removes unused interpolator inputs.

Without the **removeUnusedInputs** attribute, the HLSL compiler declares all interpolator inputs in the microcode, whether they are used or not. If interpolator inputs are not used, and the [unused](#) attribute is not applied to them, the compiler displays a warning.

The **removeUnusedInputs** attribute can sometimes improve performance for shaders that are interpolant-bound. Removing unused interpolants allows the shader to spend less time in the interpolant calculation phase.

The **removeUnusedInputs** attribute is sometimes disadvantageous, because it can cause vertex shader outputs and pixel shader inputs to be mismatched. If the two are mismatched, the shaders are patched, which can decrease performance.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

sampreg (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Sets the ranges of pixel sampler and vertex sampler registers used by the compiler.

Syntax

```
[sampreg(n,m)]
```

Parameters

n

Starting vertex sampler register.

m

Starting pixel sampler register.

Scope

Applies to functions. The **sampreg** attribute is ignored on any function except the shader entry point that is currently being compiled. The parameters also apply to any function that is compiled for this entry point, whether the function is inlined or not.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

unroll (360 HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Avoids flow-control constructs.

Syntax

```
[unroll(MaxIterationCount)]
```

Parameters

MaxIterationCount

An integer that represents the iteration count. The *MaxIterationCount* parameter is optional. If it is omitted, the HLSL compiler automatically determines the iteration count.

Scope

Applies to **for**, **while**, and **do-while** statements.

Remarks

The iteration count can be any arbitrary expression and is handled much like inline microcode options. For example, the iteration count can be a function parameter. The HLSL compiler automatically inlines the function to resolve the parameter to a constant.

Examples

The following HLSL code snippet shows how to apply the **unroll** attribute.

```
float value = 0;
[unroll(3)]
while( value <= 5 )
{
    value ++;
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

unused (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Suppresses warnings about unused shader parameters.

Syntax

```
[unused]
```

Parameters

None.

Scope

Applies to function parameters.

Remarks

The **unused** attribute is useful for shaders that use preprocessors, where the same vertex shader is used with different pixel shaders. In this case, a pixel shader may not use some of the inputs.

Examples

The following HLSL code snippet shows how to apply the **unused** attribute.

```
float4 main ( float4 coord0 : TEXCOORD0,  
[unused] float4 coord1 : TEXCOORD1 ) : COLOR  
{  
    #ifdef SHADER_2  
        return coord0 + coord1;  
    #else  
        return coord0;  
    #endif  
}
```

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

xps (HLSL)

Note

This HLSL attribute is available only when developing for the Xbox 360.

Specifies that all vertex fetch operations are done after the last [vfetch](#) instruction. This instruction is used to reduce the latency back to the command processor to free the vertex buffer.

Syntax

```
[xps]
```

Parameters

None.

Scope

Applies to functions or any scope.

See Also

Concepts

[HLSL Attributes \(Xbox 360\)](#)

[Attribute Syntax](#)

[Attribute Categories](#)

Reference

[HLSL Attributes Reference \(Xbox 360\)](#)

Zune Programming

The following topics discuss programming for the Zune platform, and provide information about Zune hardware capabilities.

In This Section

[Zune Programming Considerations](#)

Describes general programming issues to consider when creating Zune games.

[Zune HD Input Overview](#)

Provides an overview of the touch screen and accelerometer input features for the Zune HD device.

[Zune Button Mapping](#)

Describes how various Zune buttons map to the buttons on a standard Xbox 360 gamepad.

[Zune Performance](#)

Provides detailed information about performance on different Zune hardware.

[Zune Networking Overview](#)

Provides an overview of networked Zune games.

[How To: Build a Simple Networked Game on Zune](#)

Demonstrates networking functionality on Zune.

Zune Programming Considerations

Describes general programming issues to consider when creating Zune games.

Graphics

When programming for Zune, these graphics considerations are the most important:

- Zune supports 2D graphics only.
- Zune has a fixed 240x320 display size.

Zune Display Surface

The Zune device has a fixed display size of 240x320 pixels. The display can be used in either portrait or landscape orientation.

Zune games use the full 240x320 resolution of the Zune display. The entire display surface is available for text, and for critical game information such as score, number of lives, ammo, and so on. Games can draw the background texture across the entire display.

On Zune, games don't need to perform calculation of a title safe region. However, code that performs this calculation within a `#if XBOX` conditional directive (in order to limit the calculation to the Xbox 360) will continue to function correctly. See [How To: Draw a Sprite](#) for an example.

Zune supports only one aspect ratio, 4:3 (or 3:4, when held vertically). Neither the XNA Framework nor the Zune device performs any aspect ratio conversion or letterboxing.

3D Graphics

The Zune device supports 2D graphics only. It does not support 3D graphics or 3D acceleration, including 3D primitives, 3D textures, index and vertex buffers, shaders, effects, and multisampling. Zune does not implement classes that provide 3D functionality, including the following: [Texture3D](#), [IndexBuffer](#), [Model](#), [BasicEffect](#), [PixelShader](#), [ClipPlane](#), [DepthStencilBuffer](#), [Effect](#), and [SamplerState](#).

The documentation for each type and member in the XNA framework identifies supported platforms.

ResourceUsage Flags

[TextureUsage](#) flag settings other than [TextureUsage.None](#) have no effect on Zune. The XNA Framework will still raise exceptions for invalid combinations of [TextureUsage](#) flags and other texture settings (such as format and number of mipmap levels). Use [TextureUsage.None](#) for Windows, Xbox 360, and Zune platforms, whenever possible.

Input

For input, the Zune device has two buttons and either a control pad or a Zune Pad. These inputs map to controller input, retrieved with the [GamePad.GetState](#) method. For more information on specific mappings and programmatically detecting user input, see [Zune Button Mapping](#).

Note

APIs related to input devices not available on Zune, such as [Mouse](#) and [Keyboard](#), can be accessed in code running on a Zune device. The **X** and **Y** values in [MouseState](#) returned on Zune will always be zero. Similarly, checking for keys pressed using [Keyboard](#) on Zune will always return keys in an unpressed state. Also, you should not expect an exception when you access the mouse, keyboard, or other unexpected input devices on Zune.

Media and Audio

Zune audio uses the new [SoundEffect](#) and [SoundEffectInstance](#) classes for sound effects, including playing .mp3 and .wma files. Zune games use the new [Microsoft.Xna.Framework.Media](#) namespace and classes for playing songs on the Zune device and for viewing pictures in the media library. To create a rich media experience, games use the capabilities of the **Media** namespace to incorporate media from the device library directly into gameplay.

Zune does not use XACT audio wave banks or sound banks.

Storage

To get a storage device and then create containers and files on that device, Zune games use the [BeginShowStorageDeviceSelector](#) method, just as Xbox 360 games do. On Zune, this call immediately returns the [StorageDevice](#) object. [How To: Get a StorageDevice Asynchronously](#) demonstrates this process.

The maximum size of an XNA Framework project on a Zune device is 2 GB. This means the entire game, including support files such as level data, sound, and so on, must not exceed that size.

The maximum memory usage of an XNA Framework game on a Zune device is 16 MB. This means that memory usage and content loaded by the game (graphics, sound, levels, and so on) should not exceed 16 MB.

Source Code

Zune projects define a ZUNE symbol for use with conditional compilation directives such as **#if**. You can use conditional compilation to maintain source code that is shared between projects, but has different behaviors for Windows, Xbox 360, or Zune. [How To: Draw a Sprite](#) gives you an example of conditional compilation where the available screen space is calculated differently for Xbox 360 than for Windows and Zune. (See [#if \(C# Reference\)](#) and [C# Preprocessor Directives](#) for more information on conditional compilation and C# preprocessor directives.)

XNA Game Studio and Your Zune

Exiting a Game

When an XNA Framework game exits on the Zune, the Zune device reboots. This behavior is by design (and is also desirable—it ensures that game and runtime resources are completely cleaned up after a game exits).

Holding down the Back button for 2 seconds will force the currently executing game to exit, regardless of the game's normal exit semantics.

Rebooting Your Zune

If your Zune device becomes unresponsive or is not operating as expected, you may want to reboot it.

To restart the Zune device

1. Set the **Hold** switch to the unlocked position.

Note

To make sure that the **Hold** switch is in the unlocked position, you may have to move the switch first to the locked position, and then to the unlocked position. This ensures the switch is fully in the unlocked position.

2. Press and hold the **Back** button as you press the top portion ('**Up**') on the **Zune Pad**.
3. Release both buttons when the Zune device begins to restart.

Typically, the device will restart 2-3 seconds after you begin holding both buttons down.

For additional Zune device information, visit www.zune.net.

See Also

Concepts

[How To: Draw a Sprite](#)

[Connecting to your Zune Device with XNA Game Studio](#)

Reference

[SoundEffect Class](#)

[SoundEffectInstance Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

[GamePadState.DPad Property](#)

[GamePadState.Buttons Property](#)

[#if \(C# Reference\)](#)

[C# Preprocessor Directives](#)

Zune HD Input Overview

Provides an overview of the touch screen and accelerometer input features for the Zune HD device.

XNA Game Studio provides support for developing applications that use input gathered from the built-in touch screen and accelerometer of the Zune HD device. For an example of touch screen and accelerometer programming, see [Platformer: Adding Touch Support](#).

Touch Screen

The touch screen of the Zune HD device can detect multiple touches and the pressure of each touch. The touch screen can track up to four separate touch locations at the same time. Each location tracks values such as screen location, pressure of the touch, and the state of the touch (such as pressed or released). These touch locations and their information is stored in a [TouchCollection](#) structure.

The following tables demonstrate common scenarios of touch interactions with the device, and the resulting values of the touch location collection.

The easiest case is when the screen is not touched. The value of the [Count](#) property is zero, and the [TouchCollection](#) object has no additional information.

The following table is generated when a finger taps the screen:

Frame	Touch Collection ([item1], [item2], ...)
0	(([Id = 1, State = Pressed])
1	(([Id = 1, State = Released])
2	N/A

The following table is generated when a finger touches the screen, moves across the screen, and then is lifted:

Frame	Touch Collection ([item1], [item2], ...)
0	(([Id = 1, State = Pressed])
1	(([Id = 1, State = Moved])
2	(([Id = 1, State = Moved])
...	(([Id = 1, State = Moved])
n	(([Id = 1, State = Released])
n+1	N/A

The following table is generated when two fingers simultaneously touch the screen, move across the screen, and then each finger is lifted individually:

Frame	Touch Collection ([item1], [item2], ...)
0	(([Id = 1, State = Pressed], [Id = 2, State = Pressed])
1	(([Id = 1, State = Moved], [Id = 2, State = Moved])
2	(([Id = 1, State = Moved], [Id = 2, State = Moved])
3	(([Id = 1, State = Released], [Id = 2, State = Moved])
4	(([Id = 2, State = Moved])
5	(([Id = 2, State = Released])
6	N/A

Basic Usage Code for the Touch Screen API

The following code sample calls the `DrawStroke` method whenever a stroke gesture is detected:

C#

```
// Process touch locations.
foreach (TouchLocation location in curTouches)
{
    switch (location.State)
    {
        case TouchLocationState.Pressed:
            //Start tracking a particular touch location
            //In this example, you start the stroke from a special
            //area of the screen.
```

```

        if (castingArea.Contains((int)(location.Position.X),
            (int)(location.Position.Y)))
        {
            magicSpellStrokeId = location.Id;
        }
        break;
    case TouchLocationState.Moved:
        TouchLocation prevLoc;
        if (location.TryGetPreviousLocation(out prevLoc))
            DrawStroke(prevLoc.Position, location.Position);
        break;
    case TouchLocationState.Released:
        if (strokeBeginning)
        {
            if (location.Id == magicSpellStrokeId)
                magicSpellStrokeId = -1;
            curStr = waitingStr;
        }
        break;
    }
}

```

Tip

In the sample above, the ID of a specific touch location is stored by the `magicSpellStrokeId` variable. If this variable is declared at a higher level, it becomes very easy to track that touch location over multiple frames. This is because a touch location never changes its ID, but its index in the touch collection can change.

For more information about using the touch screen API in your application, see [Platformer: Adding Touch Support](#).

Accelerometer

The Zune HD device has a built-in accelerometer that detects acceleration along three axes: x, y, and z. This mirrors the right-handed coordinate system used by the XNA framework:

- Positive x-axis pointing to the right.
- Positive y-axis pointing upwards.
- Positive z-axis pointing out from the front of the device (towards the observer).

This acceleration, measured as G-force (g), ranges from -1 to 1 for each axis.

For example, when the device is held upright, aligned along the y axis, the acceleration value is (0, -1, 0). The x- and z-axes values are zero because the device isn't moving along those lines, and the y-axis value is -1 due to the effect of Earth's gravity, relative to the device. Therefore, to achieve an acceleration value of (0, 0, 0), the device would have to be dropping along the y-axis at a rate of 1 g. This cancels the omnipresent effect of the Earth's gravity.

You can use this acceleration vector (for example, (0, -1, 0)) to determine the current tilt of the device with simple trigonometry. For example, when the device is tilted 45 degrees around the z-axis (towards the right), the acceleration vector is (0.5, -0.5, 0). Using this value, the z-axis rotation can be computed using the rotation matrix returned by the [GetRotation](#) method and the following formula: `float zRot = (float)Math.Atan2(-rotationMatrix.Right.Y, rotationMatrix.Right.X);`. The result is -.8 radians (or -45 degrees).

You can use this approach for the remaining axes. The resultant code is as follows:

C#

```

Matrix rotation = acc1State.GetRotation();
float xrot =
    (float)Math.Atan2(rotation.Forward.Y, rotation.Forward.Z);
float yrot = (float)Math.Atan2(rotation.Up.X, rotation.Up.Z);
float zrot = (float)Math.Atan2(-rotation.Right.Y, rotation.Right.X);

```

Note

Since the Zune HD device does not have a gyroscope sensor, this tilt computation is not always accurate. For example, the tilt angle changes when you hold the device in the same orientation and move the device along a directional axis.

Basic Usage Code for the Accelerometer API

The following code is a simple example of using accelerometer data to alter a screen position. Tilting the device causes the position to shift in the direction the device is tilted:

C#

```
// Get touch state.  
acclState = Accelerometer.GetState();  
//Update position based on the tilt of the device  
position.X += acclState.Acceleration.X * 10.0f;
```

For more information about using the accelerometer API in your application, see [Platformer: Adding Touch Support](#).

See Also

Concepts

Reference

[Accelerometer Class](#)

[TouchPanel Class](#)

Zune Button Mapping

Describes how various Zune buttons map to the buttons on a standard Xbox 360 gamepad.

Input

For input, the Zune device has two buttons and either a device control pad (version 1.0) or a Zune Pad (version 2.0). These inputs map to controller input, retrieved with the [GamePad.GetState](#) method.

Back and Play/Pause Buttons

The **Back** button maps to [GamePadState.Buttons.Back](#), and the **Play/pause** button maps to [GamePadState.Buttons.B](#).



Device Control Pad (version 1.0)

The control pad provides input for presses (or 'clicks') on the pad.

Directional presses on the control pad map to the [GamePadState.DPad](#) directions: up, down, left, and right, or to a combination of two of those directions. Pressing straight down in the middle of the control pad maps to [GamePadState.Buttons.A](#).



Zune Pad

In addition to the behavior described in the previous sections, touch or motion over the Zune Pad is also reported as left thumbstick position in the [GamePadState.ThumbSticks.Left](#) property. Finally, clicking anywhere on the Zune Pad maps to [GamePadState.Buttons.LeftShoulder](#).



See Also

Concepts

[Connecting to your Zune Device with XNA Game Studio](#)

Reference

[GamePadState.DPad Property](#)

[GamePadState.Buttons Property](#)

Zune Performance

Provides detailed information about performance on different Zune hardware.

Screen Refresh Rates

	Refresh Rate	Screen Resolution
Zune 30	60 Hertz	240 x 320
Zune 4, 8, or 80	30 Hertz	240 x 320

Developers may notice performance differences between the Zune 30, and the Zune 4, 8, and 80. While the hardware of the two devices is similar, screen refresh rates are more frequent on the Zune 30. In terms of actual performance, however, the difference between the Zune 30, on the one hand, and the Zune 4, 8, and 80, on the other, is nearly imperceptible.

The screen resolution on all devices is exactly the same: 240 x 320. On Zune 30, the screen refreshes at 60 Hertz or 60 times a second. On Zune 4, 8, and 80, the screen refreshes at 30 Hertz or 30 times a second. The less-frequent refresh rate saves battery life on the Zune 4, 8, and 80. Prolonging battery life is a major benefit for users.

Internally, the XNA Framework is optimized to save Zune battery life. A call to [Game.Draw Method](#) is not worthwhile unless the screen can refresh. For example, if the screen can only draw a sprite 30 times in one second, it would waste processing power and battery life to draw the game screen 45 times in one second. Update will still be called as much as possible regardless of the screen refresh.

Code that calculates frames per second (FPS) in a Draw loop will show close to 60 FPS on a Zune 30. That same code will show close to 30 FPS on a Zune 80. This difference provides savings in battery life, a critical performance metric on a mobile device.

Battery Life

Processors in all versions of the Zune device are similar enough to be considered virtually identical. However, the processor in the Zune 4, 8, and 80 runs at 399 MHz, and the processor in the Zune 30 runs at 524 MHz. This difference gives the new series of Zune devices more battery life.

See Also

Concepts

[Zune Programming Considerations](#)

Zune Networking Overview

Provides an overview of networked Zune games.

Networking for Zune is designed to be consistent with Xbox 360 and Windows. The general structure of a networked game on Zune will be similar to a networked game on Xbox 360 or Windows.

Zune Networking APIs

Two XNA namespaces support Zune networking. The [Microsoft.Xna.Framework.Net](#) namespace provides actual networking functionality. For example, it supports APIs for network creation and network related events. The [Microsoft.Xna.Framework.GamerServices](#) namespace provides access to the [Guide](#) class.

Most APIs under the [Microsoft.Xna.Framework.Net](#) namespace are supported on Zune. Some APIs in the [Microsoft.Xna.Framework.GamerServices](#) namespace are not supported. Zune networking capabilities are not based on Xbox LIVE. There is limited [Guide](#) class functionality on Zune, and Xbox LIVE IDs are not accessible in networked games on Zune.

Using the Guide Class on Zune

These [Guide](#) members are supported on Zune:

- [BeginShowStorageDeviceSelector](#)
- [EndShowStorageDeviceSelector](#)
- [Guide.IsScreenSaverEnabled](#)
- [Guide.IsVisible](#)

Use the [Guide](#) class to access local storage through the [BeginShowStorageDeviceSelector](#) and [EndShowStorageDeviceSelector](#) methods. Use the [Guide.IsVisible](#) property to check for the existence of guide functionality in code. This property is always **false** on Zune.

See the [Networking](#) section for detailed information and additional networking examples.

Note

See the *Supported Platforms* section on individual reference pages to check for Zune support. If Zune is not on the supported platforms list, the API is not supported. If listed, there is support for the API.

Zune Networking Specifics

- Eight is the maximum number of Zune devices that can simultaneously share and play in one game. This is less than the maximum on Xbox 360 and Windows.
- Networking APIs on Zune do not support Internet connectivity.
- Network types supported on Zune are [NetworkSessionType.SystemLink](#) and [NetworkSessionType.Local](#). When you create a network with either [Create](#) or [BeginCreate](#), you must use one of these network types.
- A networked Zune game can only be played on Zune devices. Players on a nearby Xbox 360 or Windows machine cannot join a game.

Events and Game Design

Zune networking features give developers tools to build creative and unique games. However, creating a networked game requires extra decisions. For example, will the game support just one player? Will the game allow other players to join mid-game? What happens when gamers leave the game in the middle of play? Answers to these questions will dictate how to build the basic game structure. Use the following events to build a game design.

Event Name	Description
GameEnded	Game moves from gameplay to the lobby
GamerJoined	New player joins a multiplayer session
GamerLeft	Player leaves the multiplayer session
GameStarted	Game moves from the lobby into actual gameplay
HostChanged	Session host has changed
SessionEnded	Multiplayer session ends

Troubleshooting

Make sure wireless is turned on when testing networked Zune games. If wireless is turned on, you will see a symbol above the battery icon in the bottom left corner of the display.

When the device is rebooted with wireless turned on, you may have to wait for wireless to return. The symbol above the battery icon needs to reappear before you try wireless after a reboot.

Zune Wireless Icon



To turn on wireless on Zune

1. Click the **settings** menu.
2. Under **settings**, click the **wireless** menu.
3. Turn wireless on by clicking **wireless: off**. If the menu says **wireless: on**, then wireless is already turned on.

See Also

Concepts

[Zune Programming Considerations](#)

Reference

[Guide Class](#)

[NetworkSession Class](#)

[NetworkSessionType Enumeration](#)

How To: Build a Simple Networked Game on Zune

Demonstrates networking functionality on Zune.

This example builds on the code shown in [Your First Game: Microsoft XNA Game Studio in 2D](#). The code from that sample is expanded to include some basic game play and simple networking. The following sections show you how to transform the simple example code into a networked game on Zune:

- [The Complete Sample](#)
- [Adding Networking Objects](#)
- [Creating or Joining an Existing Networked Game](#)
- [Reading and Writing Network Packets](#)
- [Using Network Events](#)
- [Encapsulating Code](#)

Note

Not all APIs are available on Zune. See the *Supported Platforms* section on individual reference pages to check for Zune support. If Zune is not on the supported platforms list, the API is not supported. If listed, there is support for the API.

Network connections between Zunes do not form instantaneously. When running this example, either creating or joining a network session will take a few moments.

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download NetworkSimpleGame_Sample.zip](#).

Adding Networking Objects

First, add necessary networking objects to the game class that derives from [Game](#).

C#

```
// networking objects
NetworkSession networkSession;
PacketWriter packetWriter = new PacketWriter();
PacketReader packetReader = new PacketReader();
```

Add a [GamerServicesComponent](#) `GamerServicesComponent` in the game constructor.

C#

```
Components.Add(new GamerServicesComponent(this));
```

Creating or Joining an Existing Networked Game

Two types of networks are supported on Zune, [NetworkSessionType.SystemLink](#) and [NetworkSessionType.Local](#). This example creates a [NetworkSessionType.SystemLink](#) network using the [Create](#) method. This example game allows players to join in the middle of game play, and it allows for a maximum of eight players.

C#

```
networkSession =
    NetworkSession.Create(NetworkSessionType.SystemLink,
        maxLocalGamers, maxGamers);
HookNetworkSessionEvents();
```

Reading and Writing Network Packets

This method updates the server on the status of the game. The code sends the number of gamers in the game, the enemy

location, and the location of each player sprite. Notice the order that the data is sent, and that the data is read back in reverse order in the method that executes on the clients.

C#

```
// send a packet that first indicates how many
// players it has data for
packetWriter.Write(networkSession.AllGamers.Count);

// send a packet that has position of enemy on server
packetWriter.Write(enemyPos);
packetWriter.Write(enemySpeed);

// send to all gamers in network
foreach (NetworkGamer gamer in networkSession.AllGamers)
{
    // look up this player
    Rodent r = gamer.Tag as Rodent;

    // write state of mouserat to the output network packet
    packetWriter.Write(r.Position);
    packetWriter.Write(r.Dead);
}

// send the combined data for all players to everyone in the session
LocalNetworkGamer server = (LocalNetworkGamer)networkSession.Host;
server.SendData(packetWriter, SendDataOptions.InOrder);
```

This method updates a local game based on data from the server. Again, note the order of data received.

C#

```
// Keep reading as long as incoming packets are available.
while (gamer.IsDataAvailable)
{
    NetworkGamer sender;

    // Read a single packet from the network.
    gamer.ReceiveData(packetReader, out sender);

    // If a player has recently joined or left, it is possible the
    // server might have sent information about a different number
    // of players than the client currently knows about. If so,
    // we will be unable to match up which data refers to which
    // player. The solution is just to ignore the packet for now:
    // this situation will resolve itself as soon as the client
    // gets the join/leave notification.
    if (networkSession.AllGamers.Count != packetReader.ReadInt32())
        continue;

    // read the position of the enemy
    enemyPos = packetReader.ReadVector2();
    enemySpeed = packetReader.ReadVector2();

    // next read the position of all current network gamers
    foreach (NetworkGamer remoteGamer in networkSession.AllGamers)
    {
        Rodent r = remoteGamer.Tag as Rodent;
        r.Position = packetReader.ReadVector2();
        r.Dead = packetReader.ReadBoolean();
    }
}
```

In this example, the sprite added to the game in [Your First Game: Microsoft XNA Game Studio in 2D](#) still moves around the screen without interruption from user input. The addition of networked play allows for each player to control one sprite. This method updates the position of the local gamer based on inputs and sends out update packets including the new local player position.

C#

```
int MaxX = graphics.GraphicsDevice.Viewport.Width - r.Width;
int MinX = 0;
int MaxY = graphics.GraphicsDevice.Viewport.Height - r.Height;
int MinY = 0;

// check pad input
if (IsPressed(Buttons.DPadRight))
{
    if (!(r.PositionX > MaxX))
    {
        r.PositionX += 1.2f; // move mouserat right
    }
}
else if (IsPressed(Buttons.DPadLeft))
{
    if (!(r.PositionX < MinX))
    {
        r.PositionX -= 1.2f; // move mouserat left
    }
}
else if (IsPressed(Buttons.DPadUp))
{
    if (!(r.PositionY < MinY))
    {
        r.PositionY -= 1.2f; // move mouserat up
    }
}
else if (IsPressed(Buttons.DPadDown))
{
    if (!(r.PositionY > MaxY))
    {
        r.PositionY += 1.2f; // move mouserat down
    }
}

// check for collison with the enemy cat, update dead property
CheckIfEaten(r);

// if not the server, send this update info to
// all other network gamers
if (!networkSession.IsHost)
{
    // write updated position and life status to network packet
    packetWriter.Write(r.Position);
    packetWriter.Write(r.Dead);

    // send to server
    gamer.SendData(packetWriter, SendDataOptions.InOrder,
        networkSession.Host);
}
```

This method updates the position of other players.

C#

```
// Keep reading as long as incoming packets are available.
while (gamer.IsDataAvailable)
{
    NetworkGamer sender;

    // Read a single packet from the network.
    gamer.ReceiveData(packetReader, out sender);

    if (!sender.IsLocal)
    {
        Rodent r = sender.Tag as Rodent;
```

```

        r.Position = packetReader.ReadVector2();
        r.Dead = packetReader.ReadBoolean();
    }
}

```

Using Network Events

This method hooks up event handlers. It is called when you create the network.

C#

```

void HookNetworkSessionEvents()
{
    networkSession.GamerJoined += GamerJoinedEventHandler;
    networkSession.SessionEnded += SessionEndedEventHandler;
}

```

These are the event handler methods.

C#

```

void GamerJoinedEventHandler(object sender, GamerJoinedEventArgs e)
{
    // new gamer, add new rodent player object
    e.Gamer.Tag = new Rodent(this);
    Components.Add(e.Gamer.Tag as Rodent);
}

```

C#

```

void SessionEndedEventHandler(object sender,
    NetworkSessionEndedEventArgs e)
{
    DrawMenuMessage("Network Session Has Ended.");
    networkSession.Dispose();
    networkSession = null;
}

```

Encapsulating Code

The original simple game example contained all functionality within one class. Encapsulating logic and data for the added sprite into in new class adds organization. Each game instance in a networked play can use the new class. The new class is derived from [DrawableGameComponent](#).

C#

```

class Rodent : Microsoft.Xna.Framework.DrawableGameComponent
{
    Texture2D rodentTexture;    // alive texture
    Texture2D deadRodentTexture; // dead texture
    SpriteBatch spriteBatch;   // spritebatch for drawing sprites
                                // to the screen

    bool dead;                 // is this rodent dead or alive
    Vector2 rodentPos;         // rodent position

    # region properties

    public bool Dead
    {
        get
        {
            return dead;
        }
        set
        {
            dead = value;
        }
    }
}

```

```

    }
}

public Vector2 Position
{
    get
    {
        return rodentPos;
    }
    set
    {
        rodentPos = value;
    }
}

public float PositionX
{
    get
    {
        return rodentPos.X;
    }
    set
    {
        rodentPos.X = value;
    }
}

public float PositionY
{
    get
    {
        return rodentPos.Y;
    }
    set
    {
        rodentPos.Y = value;
    }
}

public int Width
{
    get
    {
        return this.rodentTexture.Width;
    }
}

public int Height
{
    get
    {
        return this.rodentTexture.Height;
    }
}

#endregion

public Rodent(Game game)
    : base(game)
{
    dead = false;
    rodentPos = new Vector2(204.0f, 287.0f);
}

public override void Initialize()
{
    base.Initialize();
}

```

```

protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
    rodentTexture = Game.Content.Load<Texture2D>("mouserat");
    deadRodentTexture = Game.Content.Load<Texture2D>("deadmouserat");

    base.LoadContent();
}

protected override void UnloadContent()
{
    base.UnloadContent();
}

public override void Draw(GameTime gameTime)
{
    spriteBatch.Begin(SpriteBlendMode.AlphaBlend);
    if (dead)
    {
        spriteBatch.Draw(deadRodentTexture, rodentPos, Color.White);
    }
    else
    {
        spriteBatch.Draw(rodentTexture, rodentPos, Color.White);
    }
    spriteBatch.End();

    base.Draw(gameTime);
}
}

```

See Also

Concepts

[Your First Game: Microsoft XNA Game Studio in 2D](#)

[Zune Programming Considerations](#)

Reference

[Guide Class](#)

[NetworkSession Class](#)

[NetworkSessionType Enumeration](#)

[GamerServicesComponent Class](#)

[NetworkSession.Create Method](#)

.NET Compact Framework for Xbox 360

The XNA Framework includes the .NET Compact Framework for Xbox 360, which is a version of the .NET Compact Framework designed specifically for the Xbox 360 video game and entertainment system. The .NET Compact Framework for Xbox 360 implements a subset of the .NET Compact Framework, and has been optimized to take advantage of and expose the power of the Xbox 360. XNA Game Studio projects that target Windows-based game machines have access to the full .NET Framework, while projects that target Xbox 360 use the .NET Compact Framework for Xbox 360.

This section provides information on developing XNA Game Studio applications using the C# language and the .NET Compact Framework for Xbox 360.

In This Section

[Namespaces and Classes Supported by the .NET Compact Framework for Xbox 360](#)

Describes how to determine the namespaces, types, and members that are supported by the .NET Compact Framework for Xbox 360.

[Additions to the .NET Compact Framework for Xbox 360](#)

Identifies namespaces, types, and members unique to the .NET Compact Framework for Xbox 360.

[Thread Pools in the .NET Compact Framework for Xbox 360](#)

Discusses the use of thread pools in the .NET Compact Framework for Xbox 360.

[Language, Locales, and Cultures in the .NET Compact Framework for Xbox 360](#)

Describes the mapping of console language and locale to managed cultures in the .NET Compact Framework for Xbox 360.

[XNA Framework Remote Performance Monitor for Xbox 360](#)



Provides information on the XNA Framework Remote Performance Monitor for Xbox 360.

Namespaces and Classes Supported by the .NET Compact Framework for Xbox 360

The .NET Compact Framework for Xbox 360 implements a subset of the namespaces, types, and members available in the .NET Compact Framework. This topic discusses that subset and explains how to determine support for specific classes or members.

Determining Supported Namespaces, Types, and Members

The following methods help you determine the specific types and members that are supported by the .NET Compact Framework for Xbox 360.

- The online MSDN [.NET Framework Class Library](#) documentation has been updated to show the types and members that are supported on the Xbox 360. Version information in those topics indicates XNA Framework support. Also, in the member summary pages, members supported on Xbox 360 display the 'Supported by XNA Framework' icon  after the .NET Compact Framework icon .
- The [Namespaces, Types, and Members](#) topic lists the namespaces, types, and members that are supported by the .NET Compact Framework for Xbox 360.

While working within an Xbox 360 project, the following methods of determining supported types and methods are available.

- IntelliSense will show only types and members that are available to Xbox 360 projects. For information about IntelliSense, see [Using IntelliSense](#).
- You will receive compiler errors when you use unsupported types or members within an Xbox 360 project.

For code portability, you may want to use only types and members supported by the .NET Compact Framework for Xbox 360, even though XNA Game Studio projects for Windows have access to the entire .NET Framework. Doing so will make it easier to create single-source applications that compile and run on both Windows and Xbox 360.

Unsupported Classes

The following sections discuss a few of the classes that are not supported by the .NET Compact Framework for Xbox 360.

To determine the full set of namespaces, types, and members that are supported by the .NET Compact Framework for Xbox 360, see [Determining Supported Namespace, Types, and Members](#), preceding.

To see additional members available only in the .NET Compact Framework for Xbox 360, see [Additions to the .NET Compact Framework for Xbox 360](#)

Note

The .NET Compact Framework is itself a subset of the full .NET Framework. Those differences are not described here, but you can find information in [Differences with the .NET Framework](#).

User Interface Classes

The .NET Compact Framework for Xbox 360 does not include the [System.Windows.Forms](#) namespace or any of its classes. User interface elements in an XNA game are composed of 2D or 3D graphics (or both), created using classes in the [Microsoft.Xna.Framework.Graphics Namespace](#) namespace.

Games typically have menus and other user interface elements that are unique and customized to the game, rather than standardized versions that wouldn't adequately express the game's unique feel and flow.

Web and Network Classes

The .NET Compact Framework for Xbox 360 does not include the [System.Web](#) namespace. It supports only a small subset of the [System.Net](#) namespace. See [Determining Supported Namespace, Types, and Members](#), preceding, for information on determining that subset.

Games developed for Xbox 360 using XNA Game Studio can use the functionality provided in the [Microsoft.Xna.Framework.Net](#) namespace to create system link or Xbox LIVE multiplayer games.

See Also

Concepts

[.NET Compact Framework for Xbox 360](#)

[Namespaces, Types, and Members in the .NET Compact Framework for Xbox 360](#)

[Additions to the .NET Compact Framework for Xbox 360](#)

[Differences with the .NET Framework](#)

[Using IntelliSense](#)

Reference



[Namespaces, Types, and Members in the .NET Compact Framework for Xbox 360](#)

[.NET Framework Class Library \(MSDN Online\)](#)

Namespaces, Types, and Members in the .NET Compact Framework for Xbox 360

This topic lists all of the namespaces, types, and members supported by the .NET Compact Framework for Xbox 360.

You can use this list to browse the supported members or to search for specific members. To search, click in this topic window and then press CTRL+F.

The links for each type or member jump to the corresponding online MSDN documentation. The online MSDN [.NET Framework Class Library](#) documentation has been updated to show the types and members that are supported on the Xbox 360. Version information in those topics indicates XNA Framework support. Also, in the member summary pages, members supported on Xbox 360 display the 'Supported by XNA Framework' icon  after the .NET Compact Framework icon .

For more information, see [Namespaces and Classes Supported by the .NET Compact Framework for Xbox 360](#).

Namespaces

The following namespaces are supported in the .NET Compact Framework for Xbox 360.

[System](#)
[System.CodeDom.Compiler](#)
[System.Collections](#)
[System.Collections.Generic](#)
[System.Collections.ObjectModel](#)
[System.Collections.Specialized](#)
[System.ComponentModel](#)
[System.Configuration.Assemblies](#)
[System.Diagnostics](#)
[System.Diagnostics.CodeAnalysis](#)
[System.Globalization](#)
[System.IO](#)
[System.Net](#)
[System.Reflection](#)
[System.Resources](#)
[System.Runtime.CompilerServices](#)
[System.Runtime.InteropServices](#)
[System.Runtime.Serialization](#)
[System.Security](#)
[System.Security.Permissions](#)
[System.Security.Policy](#)
[System.Text](#)
[System.Text.RegularExpressions](#)
[System.Threading](#)
[System.Xml](#)
[System.Xml.Schema](#)
[System.Xml.Serialization](#)
[System.Xml.XPath](#)
[System.Xml.Xsl](#)

(These links jump to the corresponding namespace in the list below.)

Namespaces, Types, and Members

The following namespaces, types, and members are supported in the .NET Compact Framework for Xbox 360.

Note

To limit the size of this large list, explicit interface implementations and overloads are not listed.

Note

You can open a link in a new tab within Microsoft Document Explorer by holding down SHIFT while clicking the link.

System namespace

System.Action<T> delegate

System.Activator class

[Activator Methods]
Activator.CreateInstance

System.AppDomain class

[AppDomain Methods]
AppDomain.CreateDomain
AppDomain.ExecuteAssembly
AppDomain.ToString
AppDomain.Unload
[AppDomain Properties]
AppDomain.CurrentDomain
AppDomain.FriendlyName
[AppDomain Events]
AppDomain.UnhandledException

System.AppDomainManager class

AppDomainManager.AppDomainManager
[AppDomainManager Methods]
AppDomainManager.InitializeNewDomain
[AppDomainManager Properties]
AppDomainManager.InitializationFlags

System.AppDomainManagerInitializationOptions enumeration

System.AppDomainSetup class

AppDomainSetup.AppDomainSetup

System.AppDomainUnloadedException class

AppDomainUnloadedException.AppDomainUnloadedException

System.ApplicationException class

ApplicationException.ApplicationException

System.ArgumentException class

ArgumentException.ArgumentException

System.ArgumentNullException class

ArgumentNullException.ArgumentNullException

System.ArgumentOutOfRangeException class

ArgumentOutOfRangeException.ArgumentOutOfRangeException

System.ArithmeticException class

ArithmeticException.ArithmeticException

System.Array class

[Array Methods]
Array.BinarySearch
Array.Clear
Array.Clone
Array.Copy
Array.CopyTo
Array.CreateInstance
Array.GetEnumerator
Array.GetLength

- Array.GetLowerBound
- Array.GetUpperBound
- Array.GetValue
- Array.IndexOf
- Array.Initialize
- Array.LastIndexOf
- Array.Resize<T>
- Array.Reverse
- Array.SetValue
- Array.Sort
- Array.TrueForAll<T>
- [Array Properties]
- Array.IsFixedSize
- Array.IsReadOnly
- Array.IsSynchronized
- Array.Length
- Array.Rank
- Array.SyncRoot

System.ArraySegment<T> structure

- ArraySegment<T>.ArraySegment<T>
- [ArraySegment<T> Methods]
- ArraySegment<T>.Equals
- ArraySegment<T>.GetHashCode
- ArraySegment<T>.op_Equality
- ArraySegment<T>.op_Inequality
- [ArraySegment<T> Properties]
- ArraySegment<T>.Array
- ArraySegment<T>.Count
- ArraySegment<T>.Offset

System.ArrayTypeMismatchException class

- ArrayTypeMismatchException.ArrayTypeMismatchException

System.AsyncCallback delegate

System.Attribute class

- Attribute.Attribute
- [Attribute Methods]
- Attribute.Equals
- Attribute.GetCustomAttribute
- Attribute.GetCustomAttributes
- Attribute.GetHashCode
- Attribute.IsDefined
- Attribute.Match

System.AttributeTargets enumeration

System.AttributeUsageAttribute class

- AttributeUsageAttribute.AttributeUsageAttribute
- [AttributeUsageAttribute Properties]
- AttributeUsageAttribute.AllowMultiple
- AttributeUsageAttribute.Inherited
- AttributeUsageAttribute.ValidOn

System.BadImageFormatException class

- BadImageFormatException.BadImageFormatException

System.BitConverter class

- [BitConverter Fields]
- BitConverter.IsLittleEndian
- [BitConverter Methods]

- BitConverter.GetBytes
- BitConverter.ToBoolean
- BitConverter.ToChar
- BitConverter.ToDouble
- BitConverter.ToInt16
- BitConverter.ToInt32
- BitConverter.ToInt64
- BitConverter.ToSingle
- BitConverter.ToString
- BitConverter.ToUInt16
- BitConverter.ToUInt32
- BitConverter.ToUInt64

System.Boolean structure

[Boolean Fields]

- Boolean.FalseString
- Boolean.TrueString

[Boolean Methods]

- Boolean.CompareTo
- Boolean.Equals
- Boolean.GetHashCode
- Boolean.GetTypeCode
- Boolean.Parse
- Boolean.ToString

System.Buffer class

[Buffer Methods]

- Buffer.BlockCopy
- Buffer.ByteLength
- Buffer.GetByte
- Buffer.SetByte

System.Byte structure

[Byte Fields]

- Byte.MaxValue
- Byte.MinValue

[Byte Methods]

- Byte.CompareTo
- Byte.Equals
- Byte.GetHashCode
- Byte.GetTypeCode
- Byte.Parse
- Byte.ToString

System.CannotUnloadAppDomainException class

CannotUnloadAppDomainException.CannotUnloadAppDomainException

System.Char structure

[Char Fields]

- Char.MaxValue
- Char.MinValue

[Char Methods]

- Char.CompareTo
- Char.Equals
- Char.GetHashCode
- Char.GetNumericValue
- Char.GetTypeCode
- Char.GetUnicodeCategory
- Char.IsControl
- Char.IsDigit
- Char.IsLetter
- Char.IsLetterOrDigit

- Char.IsLower
- Char.IsNumber
- Char.IsPunctuation
- Char.IsSeparator
- Char.IsSurrogate
- Char.IsSurrogatePair
- Char.IsSymbol
- Char.IsUpper
- Char.IsWhiteSpace
- Char.ToLower
- Char.ToLowerInvariant
- Char.ToString
- Char.ToUpper

System.CharEnumerator class

- [CharEnumerator Methods]
 - CharEnumerator.Clone
 - CharEnumerator.MoveNext
 - CharEnumerator.Reset
- [CharEnumerator Properties]
 - CharEnumerator.Current

System.CLSCompliantAttribute class

- CLSCompliantAttribute.CLSCompliantAttribute
- [CLSCompliantAttribute Properties]
 - CLSCompliantAttribute.IsCompliant

System.Comparison<T> delegate

System.Console class

- [Console Methods]
 - Console.ReadLine
 - Console.SetError
 - Console.SetIn
 - Console.SetOut
 - Console.Write
 - Console.WriteLine
- [Console Properties]
 - Console.Error
 - Console.In
 - Console.Out

System.Convert class

- [Convert Fields]
 - Convert.DBNull
- [Convert Methods]
 - Convert.ChangeType
 - Convert.FromBase64CharArray
 - Convert.FromBase64String
 - Convert.GetTypeCode
 - Convert.IsDBNull
 - Convert.ToBase64CharArray
 - Convert.ToBase64String
 - Convert.ToBoolean
 - Convert.ToByte
 - Convert.ToChar
 - Convert.ToDateTime
 - Convert.ToDecimal
 - Convert.ToDouble
 - Convert.ToInt16
 - Convert.ToInt32
 - Convert.ToInt64

- Convert.ToSByte
- Convert.ToSingle
- Convert.ToString
- Convert.ToUInt16
- Convert.ToUInt32
- Convert.ToUInt64

System.Converter<TInput><TOutput> delegate

System.DateTime structure

DateTime.DateTime

[DateTime Fields]

DateTime.MaxValue

DateTime.MinValue

[DateTime Methods]

DateTime.Add

DateTime.AddDays

DateTime.AddHours

DateTime.AddMilliseconds

DateTime.AddMinutes

DateTime.AddMonths

DateTime.AddSeconds

DateTime.AddTicks

DateTime.AddYears

DateTime.Compare

DateTime.CompareTo

DateTime.DaysInMonth

DateTime.Equals

DateTime.FromFileTime

DateTime.FromFileTimeUtc

DateTime.FromOADate

DateTime.GetDateTimeFormats

DateTime.GetHashCode

DateTime.GetTypeCode

DateTime.IsDaylightSavingTime

DateTime.IsLeapYear

DateTime.op_Addition

DateTime.op_Equality

DateTime.op_GreaterThan

DateTime.op_GreaterThanOrEqual

DateTime.op_Inequality

DateTime.op_LessThan

DateTime.op_LessThanOrEqual

DateTime.op_Subtraction

DateTime.Parse

DateTime.ParseExact

DateTime.SpecifyKind

DateTime.Subtract

DateTimeToFileTime

DateTimeToFileTimeUtc

DateTime.ToLocalTime

DateTime.ToLongDateString

DateTime.ToLongTimeString

DateTime.ToOADate

DateTime.ToShortDateString

DateTime.ToShortTimeString

DateTime.ToString

DateTime.ToUniversalTime

[DateTime Properties]

DateTime.Date

DateTime.Day

DateTime.DayOfWeek

DateTime.DayOfYear
DateTime.Hour
DateTime.Kind
DateTime.Millisecond
DateTime.Minute
DateTime.Month
DateTime.Now
DateTime.Second
DateTime.Ticks
DateTime.TimeOfDay
DateTime.Today
DateTime.UtcNow
DateTime.Year

System.DateTimeKind enumeration

System.DayOfWeek enumeration

System.DBNull class

[DBNull Fields]

DBNull.Value

[DBNull Methods]

DBNull.GetTypeCode

DBNull.ToString

System.Decimal structure

Decimal.Decimal

[Decimal Fields]

Decimal.MaxValue

Decimal.MinusOne

Decimal.MinValue

Decimal.One

Decimal.Zero

[Decimal Methods]

Decimal.Add

Decimal.Compare

Decimal.CompareTo

Decimal.Divide

Decimal.Equals

Decimal.Floor

Decimal.GetBits

Decimal.GetHashCode

Decimal.GetTypeCode

Decimal.Multiply

Decimal.Negate

Decimal.op_Addition

Decimal.op_Decrement

Decimal.op_Division

Decimal.op_Equality

Decimal.op_Explicit

Decimal.op_GreaterThan

Decimal.op_GreaterThanOrEqual

Decimal.op_Implicit

Decimal.op_Increment

Decimal.op_Inequality

Decimal.op_LessThan

Decimal.op_LessThanOrEqual

Decimal.op_Modulus

Decimal.op_Multiply

Decimal.op_Subtraction

Decimal.op_UnaryNegation

Decimal.op_UnaryPlus

- Decimal.Parse
- Decimal.Remainder
- Decimal.Round
- Decimal.Subtract
- Decimal.ToByte
- Decimal.ToDouble
- Decimal.ToInt16
- Decimal.ToInt32
- Decimal.ToInt64
- Decimal.ToSByte
- Decimal.ToSingle
- Decimal.ToString
- Decimal.ToUInt16
- Decimal.ToUInt32
- Decimal.ToUInt64
- Decimal.Truncate

System.Delegate class

[Delegate Methods]

- Delegate.Clone
- Delegate.Combine
- Delegate.CombineImpl
- Delegate.Equals
- Delegate.Finalize
- Delegate.GetHashCode
- Delegate.GetInvocationList
- Delegate.op_Equality
- Delegate.op_Inequality
- Delegate.Remove
- Delegate.RemoveImpl

System.DivideByZeroException class

DivideByZeroException.DivideByZeroException

System.Double structure

[Double Fields]

- Double.Epsilon
- Double.MaxValue
- Double.MinValue
- Double.NaN
- Double.NegativeInfinity
- Double.PositiveInfinity

[Double Methods]

- Double.CompareTo
- Double.Equals
- Double.GetHashCode
- Double.GetTypeCode
- Double.IsInfinity
- Double.IsNaN
- Double.IsNegativeInfinity
- Double.IsPositiveInfinity
- Double.Parse
- Double.ToString

System.EntryPointNotFoundException class

EntryPointNotFoundException.EntryPointNotFoundException

System.Enum structure

Enum.Enum

[Enum Methods]

- Enum.CompareTo
- Enum.Equals

- Enum.GetHashCode
- Enum.GetTypeCode
- Enum.GetUnderlyingType
- Enum.IsDefined
- Enum.Parse
- Enum.ToObject
- Enum.ToString

System.Environment class

[Environment Methods]

- Environment.GetFolderPath

[Environment Properties]

- Environment.OSVersion
- Environment.ProcessorCount
- Environment.TickCount
- Environment.Version

System.Environment.SpecialFolder enumeration

System.EventArgs class

- EventArgs.EventArgs

[EventArgs Fields]

- EventArgs.Empty

System.EventHandler delegate

System.EventHandler<TEventArgs> delegate

System.Exception class

- Exception.Exception

[Exception Methods]

- Exception.GetBaseException

- Exception.ToString

[Exception Properties]

- Exception.HResult
- Exception.InnerException
- Exception.Message
- Exception.StackTrace

System.FlagsAttribute class

- FlagsAttribute.FlagsAttribute

System.FormatException class

- FormatException.FormatException

System.GC class

[GC Methods]

- GC.Collect

- GC.GetTotalMemory

- GC.KeepAlive

- GC.ReRegisterForFinalize

- GC.SuppressFinalize

- GC.WaitForPendingFinalizers

[GC Properties]

- GC.MaxGeneration

System.Guid structure

- Guid.Guid

[Guid Fields]

- Guid.Empty

[Guid Methods]

- Guid.CompareTo

Guid.Equals
Guid.GetHashCode
Guid.NewGuid
Guid.op_Equality
Guid.op_Inequality
Guid.ToArray
Guid.ToString

System.IAsyncResult interface

[IAsyncResult Properties]
IAsyncResult.AsyncState
IAsyncResult.AsyncWaitHandle
IAsyncResult.CompletedSynchronously
IAsyncResult.IsCompleted

System.ICloneable interface

[ICloneable Methods]
ICloneable.Clone

System.IComparable interface

[IComparable Methods]
IComparable.CompareTo

System.IComparable<T> interface

[IComparable<T> Methods]
IComparable<T>.CompareTo

System.IConvertible interface

[IConvertible Methods]
IConvertible.GetTypeCode
IConvertible.ToBoolean
IConvertible.ToByte
IConvertible.ToChar
IConvertible.ToDateTime
IConvertible.ToDecimal
IConvertible.ToDouble
IConvertible.ToInt16
IConvertible.ToInt32
IConvertible.ToInt64
IConvertible.ToSByte
IConvertible.ToSingle
IConvertible.ToString
IConvertible.ToType
IConvertible.ToUInt16
IConvertible.ToUInt32
IConvertible.ToUInt64

System.ICustomFormatter interface

[ICustomFormatter Methods]
ICustomFormatter.Format

System.IDisposable interface

[IDisposable Methods]
IDisposable.Dispose

System.IEquatable<T> interface

[IEquatable<T> Methods]
IEquatable<T>.Equals

System.IFormatProvider interface

[IFormatProvider Methods]
IFormatProvider.GetFormat

System.IFormattable interface

[IFormattable Methods]

IFormattable.ToString

System.IndexOutOfRangeException class

IndexOutOfRangeException.IndexOutOfRangeException

System.Int16 structure

[Int16 Fields]

Int16.MaxValue

Int16.MinValue

[Int16 Methods]

Int16.CompareTo

Int16.Equals

Int16.GetHashCode

Int16.GetTypeCode

Int16.Parse

Int16.ToString

System.Int32 structure

[Int32 Fields]

Int32.MaxValue

Int32.MinValue

[Int32 Methods]

Int32.CompareTo

Int32.Equals

Int32.GetHashCode

Int32.GetTypeCode

Int32.Parse

Int32.ToString

System.Int64 structure

[Int64 Fields]

Int64.MaxValue

Int64.MinValue

[Int64 Methods]

Int64.CompareTo

Int64.Equals

Int64.GetHashCode

Int64.GetTypeCode

Int64.Parse

Int64.ToString

System.IntPtr structure

IntPtr.IntPtr

[IntPtr Fields]

IntPtr.Zero

[IntPtr Methods]

IntPtr.Equals

IntPtr.GetHashCode

IntPtr.op_Equality

IntPtr.op_Explicit

IntPtr.op_Inequality

IntPtr.ToInt32

IntPtr.ToInt64

IntPtr.ToPointer

IntPtr.ToString

[IntPtr Properties]

IntPtr.Size

System.InvalidCastException class

[InvalidCastException.InvalidCastException](#)

System.InvalidOperationException class
[InvalidOperationException.InvalidOperationException](#)

System.InvalidProgramException class
[InvalidProgramException.InvalidProgramException](#)

System.IServiceProvider interface
[IServiceProvider Methods]
[IServiceProvider.GetService](#)

System.LocalDataStoreSlot class

System.MarshalByRefObject class
[MarshalByRefObject.MarshalByRefObject](#)

System.Math class

[Math Fields]

[Math.E](#)

[Math.PI](#)

[Math Methods]

[Math.Abs](#)

[Math.Acos](#)

[Math.Asin](#)

[Math.Atan](#)

[Math.Atan2](#)

[Math.Ceiling](#)

[Math.Cos](#)

[Math.Cosh](#)

[Math.Exp](#)

[Math.Floor](#)

[Math.IEEERemainder](#)

[Math.Log](#)

[Math.Log10](#)

[Math.Max](#)

[Math.Min](#)

[Math.Pow](#)

[Math.Round](#)

[Math.Sign](#)

[Math.Sin](#)

[Math.Sinh](#)

[Math.Sqrt](#)

[Math.Tan](#)

[Math.Tanh](#)

System.MemberAccessException class
[MemberAccessException.MemberAccessException](#)

System.MethodAccessException class
[MethodAccessException.MethodAccessException](#)

System.MissingMemberException class
[MissingMemberException.MissingMemberException](#)

System.MissingMethodException class
[MissingMethodException.MissingMethodException](#)

System.MulticastDelegate class

[MulticastDelegate Methods]

[MulticastDelegate.CombineImpl](#)

[MulticastDelegate.Equals](#)

- MulticastDelegate.GetHashCode
- MulticastDelegate.GetInvocationList
- MulticastDelegate.op_Equality
- MulticastDelegate.op_Inequality
- MulticastDelegate.RemoveImpl

System.MulticastNotSupportedException class
MulticastNotSupportedException.MulticastNotSupportedException

System.NotImplementedException class
NotImplementedException.NotImplementedException

System.NotSupportedException class
NotSupportedException.NotSupportedException

System.Nullable class
[Nullable Methods]
Nullable.Compare<T>
Nullable.Equals<T>
Nullable.GetUnderlyingType

System.Nullable<T> structure
Nullable<T>.Nullable<T>
[Nullable<T> Methods]
Nullable<T>.Equals
Nullable<T>.GetHashCode
Nullable<T>.GetValueOrDefault
Nullable<T>.op_Explicit
Nullable<T>.op_Implicit
Nullable<T>.ToString
[Nullable<T> Properties]
Nullable<T>.HasValue
Nullable<T>.Value

System.NullReferenceException class
NullReferenceException.NullReferenceException

System.Object class
Object.Object
[Object Methods]
Object.Equals
Object.Finalize
Object.GetHashCode
Object.GetType
Object.MemberwiseClone
Object.ReferenceEquals
Object.ToString

System.ObjectDisposedException class
ObjectDisposedException.ObjectDisposedException
[ObjectDisposedException Properties]
ObjectDisposedException.ObjectName

System.ObsoleteAttribute class
ObsoleteAttribute.ObsoleteAttribute
[ObsoleteAttribute Properties]
ObsoleteAttribute.IsError
ObsoleteAttribute.Message

System.OperatingSystem class
OperatingSystem.OperatingSystem
[OperatingSystem Methods]

- OperatingSystem.Clone
- OperatingSystem.ToString
- [OperatingSystem Properties]
- OperatingSystem.Platform
- OperatingSystem.Version

System.OutOfMemoryException class
OutOfMemoryException.OutOfMemoryException

System.OverflowException class
OverflowException.OverflowException

System.ParamArrayAttribute class
ParamArrayAttribute.ParamArrayAttribute

System.PlatformID enumeration

System.PlatformNotSupportedException class
PlatformNotSupportedException.PlatformNotSupportedException

System.Predicate<T> delegate

System.Random class
Random.Random
[Random Methods]
Random.Next
Random.NextBytes
Random.NextDouble
Random.Sample

System.RankException class
RankException.RankException

System.RuntimeFieldHandle structure
[RuntimeFieldHandle Properties]
RuntimeFieldHandle.Value

System.RuntimeMethodHandle structure
[RuntimeMethodHandle Properties]
RuntimeMethodHandle.Value

System.RuntimeTypeHandle structure

System.SByte structure
[SByte Fields]
SByte.MaxValue
SByte.MinValue
[SByte Methods]
SByte.CompareTo
SByte.Equals
SByte.GetHashCode
SByte.GetTypeCode
SByte.Parse
SByte.ToString

System.SerializableAttribute class
SerializableAttribute.SerializableAttribute

System.Single structure
[Single Fields]
Single.Epsilon
Single.MaxValue

- Single.MinValue
- Single.NaN
- Single.NegativeInfinity
- Single.PositiveInfinity
- [Single Methods]
- Single.CompareTo
- Single.Equals
- Single.GetHashCode
- Single.GetTypeCode
- Single.IsInfinity
- Single.IsNaN
- Single.IsNegativeInfinity
- Single.IsPositiveInfinity
- Single.Parse
- Single.ToString

System.StackOverflowException class

- StackOverflowException.StackOverflowException

System.String class

- String.String
- [String Fields]
- String.Empty
- [String Methods]
- String.Clone
- String.Compare
- String.CompareOrdinal
- String.CompareTo
- String.Concat
- String.Contains
- String.Copy
- String.CopyTo
- String.EndsWith
- String.Equals
- String.Format
- String.GetEnumerator
- String.GetHashCode
- String.GetTypeCode
- String.IndexOf
- String.IndexOfAny
- String.Insert
- String.Intern
- String.IsInterned
- String.IsNullOrEmpty
- String.Join
- String.LastIndexOf
- String.LastIndexOfAny
- String.op_Equality
- String.op_Inequality
- String.PadLeft
- String.PadRight
- String.Remove
- String.Replace
- String.Split
- String.StartsWith
- String.Substring
- String.ToArray
- String.ToLower
- String.ToString
- String.ToUpper
- String.Trim
- String.TrimEnd

String.TrimStart
[String Properties]
String.Chars
String.Length

System.StringComparer class

StringComparer.StringComparer
[StringComparer Methods]
StringComparer.Compare
StringComparer.Create
StringComparer.Equals
StringComparer.GetHashCode
[StringComparer Properties]
StringComparer.CurrentCulture
StringComparer.CurrentCultureIgnoreCase
StringComparer.InvariantCulture
StringComparer.InvariantCultureIgnoreCase
StringComparer.Ordinal
StringComparer.OrdinalIgnoreCase

System.StringComparison enumeration

System.SystemException class

SystemException.SystemException

System.TimeoutException class

TimeoutException.TimeoutException

System.TimeSpan structure

TimeSpan.TimeSpan
[TimeSpan Fields]
TimeSpan.MaxValue
TimeSpan.MinValue
TimeSpan.TicksPerDay
TimeSpan.TicksPerHour
TimeSpan.TicksPerMillisecond
TimeSpan.TicksPerMinute
TimeSpan.TicksPerSecond
TimeSpan.Zero
[TimeSpan Methods]
TimeSpan.Add
TimeSpan.Compare
TimeSpan.CompareTo
TimeSpan.Duration
TimeSpan.Equals
TimeSpan.FromDays
TimeSpan.FromHours
TimeSpan.FromMilliseconds
TimeSpan.FromMinutes
TimeSpan.FromSeconds
TimeSpan.FromTicks
TimeSpan.GetHashCode
TimeSpan.Negate
TimeSpan.op_Addition
TimeSpan.op_Equality
TimeSpan.op_GreaterThan
TimeSpan.op_GreaterThanOrEqual
TimeSpan.op_Inequality
TimeSpan.op_LessThan
TimeSpan.op_LessThanOrEqual
TimeSpan.op_Subtraction
TimeSpan.op_UnaryNegation

- TimeSpan.Parse
- TimeSpan.Subtract
- TimeSpan.ToString
- [TimeSpan Properties]
- TimeSpan.Days
- TimeSpan.Hours
- TimeSpan.Milliseconds
- TimeSpan.Minutes
- TimeSpan.Seconds
- TimeSpan.Ticks
- TimeSpan.TotalDays
- TimeSpan.TotalHours
- TimeSpan.TotalMilliseconds
- TimeSpan.TotalMinutes
- TimeSpan.TotalSeconds

System.TimeZone class

- TimeZone.TimeZone
- [TimeZone Methods]
- TimeZone.GetDaylightChanges
- TimeZone.GetUtcOffset
- TimeZone.IsDaylightSavingTime
- TimeZone.ToLocalTime
- TimeZone.ToUniversalTime
- [TimeZone Properties]
- TimeZone.CurrentTimeZone
- TimeZone.DaylightName
- TimeZone.StandardName

System.Type class

- Type.Type
- [Type Fields]
- Type.Delimiter
- Type.Missing
- [Type Methods]
- Type.Equals
- Type.GetArrayRank
- Type.GetAttributeFlagsImpl
- Type.GetConstructor
- Type.GetConstructorImpl
- Type.GetConstructors
- Type.GetDefaultMembers
- Type.GetElementType
- Type.GetEvent
- Type.GetEvents
- Type.GetField
- Type.GetFields
- Type.GetGenericArguments
- Type.GetGenericTypeDefinition
- Type.GetHashCode
- Type.GetInterfaces
- Type.GetMember
- Type.GetMembers
- Type.GetMethod
- Type.GetMethodImpl
- Type.GetMethods
- Type.GetNestedType
- Type.GetNestedTypes
- Type.GetProperties
- Type.GetProperty
- Type.GetPropertyImpl
- Type.GetType

Type.GetTypeCode
Type.GetTypeFromHandle
Type.HasElementTypeImpl
Type.InvokeMember
Type.IsArrayImpl
Type.IsAssignableFrom
Type.IsByRefImpl
Type.IsCOMObjectImpl
Type.IsInstanceOfType
Type.IsPointerImpl
Type.IsPrimitiveImpl
Type.IsSubclassOf
Type.IsValueTypeImpl
Type.MakeGenericType
Type.ToString
[Type Properties]
Type.Assembly
Type.AssemblyQualifiedName
Type.Attributes
Type.BaseType
Type.ContainsGenericParameters
Type.DeclaringType
Type.DefaultBinder
Type.FullName
Type.HasElementType
Type.IsAbstract
Type.IsAnsiClass
Type.IsArray
Type.IsAutoClass
Type.IsAutoLayout
Type.IsByRef
Type.IsClass
Type.IsCOMObject
Type.IsEnum
Type.IsGenericParameter
Type.IsGenericType
Type.IsGenericTypeDefinition
Type.IsImport
Type.IsInterface
Type.IsNestedAssembly
Type.IsNestedFamANDAssem
Type.IsNestedFamily
Type.IsNestedFamORAssem
Type.IsNestedPrivate
Type.IsNestedPublic
Type.IsNotPublic
Type.IsPointer
Type.IsPrimitive
Type.IsPublic
Type.IsSealed
Type.IsSpecialName
Type.IsUnicodeClass
Type.IsValueType
Type.IsVisible
Type.MemberType
Type.Module
Type.Namespace
Type.ReflectedType
Type.TypeHandle
Type.UnderlyingSystemType

System.TypeCode enumeration

System.TypedReference structure

System.TypeLoadException class
TypeLoadException.TypeLoadException

System.UInt16 structure

[UInt16 Fields]
 UInt16.MaxValue
 UInt16.MinValue
[UInt16 Methods]
 UInt16.CompareTo
 UInt16.Equals
 UInt16.GetHashCode
 UInt16.GetTypeCode
 UInt16.Parse
 UInt16.ToString

System.UInt32 structure

[UInt32 Fields]
 UInt32.MaxValue
 UInt32.MinValue
[UInt32 Methods]
 UInt32.CompareTo
 UInt32.Equals
 UInt32.GetHashCode
 UInt32.GetTypeCode
 UInt32.Parse
 UInt32.ToString

System.UInt64 structure

[UInt64 Fields]
 UInt64.MaxValue
 UInt64.MinValue
[UInt64 Methods]
 UInt64.CompareTo
 UInt64.Equals
 UInt64.GetHashCode
 UInt64.GetTypeCode
 UInt64.Parse
 UInt64.ToString

System.UIntPtr structure

UIntPtr.UIntPtr
[UIntPtr Fields]
 UIntPtr.Zero
[UIntPtr Methods]
 UIntPtr.Equals
 UIntPtr.GetHashCode
 UIntPtr.op_Equality
 UIntPtr.op_Explicit
 UIntPtr.op_Inequality
 UIntPtr.ToPointer
 UIntPtr.ToString
 UIntPtr.ToUInt32
 UIntPtr.ToUInt64
[UIntPtr Properties]
 UIntPtr.Size

System.UnauthorizedAccessException class
UnauthorizedAccessException.UnauthorizedAccessException

System.UnhandledExceptionEventArgs class

UnhandledExceptionEventArgs.UnhandledExceptionEventArgs

[UnhandledExceptionEventArgs Properties]

UnhandledExceptionEventArgs.ExceptionObject

UnhandledExceptionEventArgs.IsTerminating

System.UnhandledExceptionHandler delegate

System.Uri class

Uri.Uri

[Uri Fields]

Uri.SchemeDelimiter

Uri.UriSchemeFile

Uri.UriSchemeFtp

Uri.UriSchemeGopher

Uri.UriSchemeHttp

Uri.UriSchemeHttps

Uri.UriSchemeMailto

Uri.UriSchemeNetPipe

Uri.UriSchemeNetTcp

Uri.UriSchemeNews

Uri.UriSchemeNntp

[Uri Methods]

Uri.Canonicalize

Uri.CheckHostName

Uri.CheckSchemeName

Uri.CheckSecurity

Uri.Compare

Uri.Equals

Uri.Escape

Uri.EscapeDataString

Uri.EscapeString

Uri.EscapeUriString

Uri.FromHex

Uri.GetComponents

Uri.GetHashCode

Uri.GetLeftPart

Uri.HexEscape

Uri.HexUnescape

Uri.IsBadFileSystemCharacter

Uri.IsBaseOf

Uri.IsExcludedCharacter

Uri.IsHexDigit

Uri.IsHexEncoding

Uri.IsReservedCharacter

Uri.IsWellFormedOriginalString

Uri.IsWellFormedUriString

Uri.MakeRelative

Uri.MakeRelativeUri

Uri.op_Equality

Uri.op_Inequality

Uri.Parse

Uri.ToString

Uri.TryCreate

Uri.Unescape

Uri.UnescapeDataString

[Uri Properties]

Uri.AbsolutePath

Uri.AbsoluteUri

Uri.Authority

Uri.DnsSafeHost

Uri.Fragment

- Uri.Host
- Uri.HostNameType
- Uri.IsAbsoluteUri
- Uri.IsDefaultPort
- Uri.IsFile
- Uri.IsLoopback
- Uri.IsUnc
- Uri.LocalPath
- Uri.OriginalString
- Uri.PathAndQuery
- Uri.Port
- Uri.Query
- Uri.Scheme
- Uri.Segments
- Uri.UserEscaped
- Uri.UserInfo

System.UriComponents enumeration

System.UriFormat enumeration

System.UriFormatException class
UriFormatException.UriFormatException

System.UriHostNameType enumeration

System.UriKind enumeration

System.UriPartial enumeration

System.ValueType class
ValueType.ValueType
[ValueType Methods]
ValueType.Equals
ValueType.GetHashCode

System.Version class
Version.Version
[Version Methods]
Version.Clone
Version.CompareTo
Version.Equals
Version.GetHashCode
Version.op_Equality
Version.op_GreaterThan
Version.op_GreaterThanOrEqual
Version.op_Inequality
Version.op_LessThan
Version.op_LessThanOrEqual
Version.ToString
[Version Properties]
Version.Build
Version.Major
Version.Minor
Version.Revision

System.Void structure

System.WeakReference class
WeakReference.WeakReference
[WeakReference Methods]
WeakReference.Finalize

[WeakReference Properties]
WeakReference.IsAlive
WeakReference.Target
WeakReference.TrackResurrection

System.CodeDom.Compiler namespace

System.CodeDom.Compiler.GeneratedCodeAttribute class

GeneratedCodeAttribute.GeneratedCodeAttribute
[GeneratedCodeAttribute Properties]
GeneratedCodeAttribute.Tool
GeneratedCodeAttribute.Version

System.Collections namespace

System.Collections.ArrayList class

ArrayList.ArrayList
[ArrayList Methods]
ArrayList.Add
ArrayList.AddRange
ArrayList.BinarySearch
ArrayList.Clear
ArrayList.Clone
ArrayList.Contains
ArrayList.CopyTo
ArrayList.GetEnumerator
ArrayList.IndexOf
ArrayList.Insert
ArrayList.InsertRange
ArrayList.Remove
ArrayList.RemoveAt
ArrayList.RemoveRange
ArrayList.Reverse
ArrayList.Sort
ArrayList.Synchronized
ArrayList.ToArray
ArrayList.TrimToSize
[ArrayList Properties]
ArrayList.Capacity
ArrayList.Count
ArrayList.IsFixedSize
ArrayList.IsReadOnly
ArrayList.IsSynchronized
ArrayList.Item
ArrayList.SyncRoot

System.Collections.BitArray class

BitArray.BitArray
[BitArray Methods]
BitArray.And
BitArray.Clone
BitArray.CopyTo
BitArray.Get
BitArray.GetEnumerator
BitArray.Not
BitArray.Or
BitArray.Set
BitArray.SetAll
BitArray.Xor
[BitArray Properties]

- BitArray.Count
- BitArray.IsReadOnly
- BitArray.IsSynchronized
- BitArray.Item
- BitArray.Length
- BitArray.SyncRoot

System.Collections.CaseInsensitiveComparer class

- CaseInsensitiveComparer.CaseInsensitiveComparer
- [CaseInsensitiveComparer Methods]
- CaseInsensitiveComparer.Compare
- [CaseInsensitiveComparer Properties]
- CaseInsensitiveComparer.Default
- CaseInsensitiveComparer.DefaultInvariant

System.Collections.CaseInsensitiveHashCodeProvider class

- CaseInsensitiveHashCodeProvider.CaseInsensitiveHashCodeProvider
- [CaseInsensitiveHashCodeProvider Methods]
- CaseInsensitiveHashCodeProvider.GetHashCode
- [CaseInsensitiveHashCodeProvider Properties]
- CaseInsensitiveHashCodeProvider.Default
- CaseInsensitiveHashCodeProvider.DefaultInvariant

System.Collections.CollectionBase class

- CollectionBase.CollectionBase
- [CollectionBase Methods]
- CollectionBase.Clear
- CollectionBase.GetEnumerator
- CollectionBase.OnClear
- CollectionBase.OnClearComplete
- CollectionBase.OnInsert
- CollectionBase.OnInsertComplete
- CollectionBase.OnRemove
- CollectionBase.OnRemoveComplete
- CollectionBase.OnSet
- CollectionBase.OnSetComplete
- CollectionBase.OnValidate
- CollectionBase.RemoveAt
- [CollectionBase Properties]
- CollectionBase.Capacity
- CollectionBase.Count
- CollectionBase.InnerList
- CollectionBase.List

System.Collections.Comparer class

- Comparer.Comparer
- [Comparer Fields]
- Comparer.Default
- Comparer.DefaultInvariant
- [Comparer Methods]
- Comparer.Compare

System.Collections.DictionaryEntry structure

- DictionaryEntry.DictionaryEntry
- [DictionaryEntry Properties]
- DictionaryEntry.Key
- DictionaryEntry.Value

System.Collections.Hashtable class

- Hashtable.Hashtable
- [Hashtable Methods]
- Hashtable.Add

- Hashtable.Clear
- Hashtable.Clone
- Hashtable.Contains
- Hashtable.ContainsKey
- Hashtable.ContainsValue
- Hashtable.CopyTo
- Hashtable.GetEnumerator
- Hashtable.GetHash
- Hashtable.KeyEquals
- Hashtable.Remove
- Hashtable.Synchronized
- [Hashtable Properties]
- Hashtable.comparer
- Hashtable.Count
- Hashtable.EqualityComparer
- Hashtable.hcp
- Hashtable.IsFixedSize
- Hashtable.IsReadOnly
- Hashtable.IsSynchronized
- Hashtable.Item
- Hashtable.Keys
- Hashtable.SyncRoot
- Hashtable.Values

System.Collections.ICollection interface

- [ICollection Methods]
- ICollection.CopyTo
- [ICollection Properties]
- ICollection.Count
- ICollection.IsSynchronized
- ICollection.SyncRoot

System.Collections.IComparer interface

- [IComparer Methods]
- IComparer.Compare

System.Collections.IDictionary interface

- [IDictionary Methods]
- IDictionary.Add
- IDictionary.Clear
- IDictionary.Contains
- IDictionary.GetEnumerator
- IDictionary.Remove
- [IDictionary Properties]
- IDictionary.IsFixedSize
- IDictionary.IsReadOnly
- IDictionary.Item
- IDictionary.Keys
- IDictionary.Values

System.Collections.IDictionaryEnumerator interface

- [IDictionaryEnumerator Properties]
- IDictionaryEnumerator.Entry
- IDictionaryEnumerator.Key
- IDictionaryEnumerator.Value

System.Collections.IEnumerable interface

- [IEnumerable Methods]
- IEnumerable.GetEnumerator

System.Collections.IEnumerator interface

- [IEnumerator Methods]

- IEnumerator.MoveNext
- IEnumerator.Reset
- [IEnumerator Properties]
- IEnumerator.Current

System.Collections.IEqualityComparer interface

- [IEqualityComparer Methods]
- IEqualityComparer.Equals
- IEqualityComparer.GetHashCode

System.Collections.IHashCodeProvider interface

- [IHashCodeProvider Methods]
- IHashCodeProvider.GetHashCode

System.Collections.IList interface

- [IList Methods]
- IList.Add
- IList.Clear
- IList.Contains
- IList.IndexOf
- IList.Insert
- IList.Remove
- IList.RemoveAt
- [IList Properties]
- IList.IsFixedSize
- IList.IsReadOnly
- IList.Item

System.Collections.Queue class

- Queue.Queue
- [Queue Methods]
- Queue.Clear
- Queue.Clone
- Queue.Contains
- Queue.CopyTo
- Queue.Dequeue
- Queue.Enqueue
- Queue.GetEnumerator
- Queue.Peek
- Queue.ToArray
- Queue.TrimToSize
- [Queue Properties]
- Queue.Count
- Queue.IsSynchronized
- Queue.SyncRoot

System.Collections.SortedList class

- SortedList.SortedList
- [SortedList Methods]
- SortedList.Add
- SortedList.Clear
- SortedList.Clone
- SortedList.Contains
- SortedList.ContainsKey
- SortedList.ContainsValue
- SortedList.CopyTo
- SortedList.GetByIndex
- SortedList.GetEnumerator
- SortedList.GetKey
- SortedList.GetKeyList
- SortedList.GetValueList
- SortedList.IndexOfKey

- SortedList.IndexOfValue
- SortedList.Remove
- SortedList.RemoveAt
- SortedList.SetByIndex
- SortedList.Synchronized
- SortedList.TrimToSize
- [SortedList Properties]
- SortedList.Capacity
- SortedList.Count
- SortedList.IsFixedSize
- SortedList.IsReadOnly
- SortedList.IsSynchronized
- SortedList.Item
- SortedList.Keys
- SortedList.SyncRoot
- SortedList.Values

System.Collections.Stack class

- Stack.Stack
- [Stack Methods]
- Stack.Clear
- Stack.Clone
- Stack.Contains
- Stack.CopyTo
- Stack.GetEnumerator
- Stack.Peek
- Stack.Pop
- Stack.Push
- Stack.Synchronized
- Stack.ToArray
- [Stack Properties]
- Stack.Count
- Stack.IsSynchronized
- Stack.SyncRoot

System.Collections.Generic namespace

System.Collections.Generic.Comparer<T> class

- Comparer<T>.Comparer<T>
- [Comparer<T> Methods]
- Comparer<T>.Compare
- [Comparer<T> Properties]
- Comparer<T>.Default

System.Collections.Generic.Dictionary<TKey><TValue> class

- Dictionary<TKey><TValue>.Dictionary<TKey><TValue>
- [Dictionary<TKey><TValue> Methods]
- Dictionary<TKey><TValue>.Add
- Dictionary<TKey><TValue>.Clear
- Dictionary<TKey><TValue>.ContainsKey
- Dictionary<TKey><TValue>.ContainsValue
- Dictionary<TKey><TValue>.GetEnumerator
- Dictionary<TKey><TValue>.Remove
- Dictionary<TKey><TValue>.TryGetValue
- [Dictionary<TKey><TValue> Properties]
- Dictionary<TKey><TValue>.Comparer
- Dictionary<TKey><TValue>.Count
- Dictionary<TKey><TValue>.Item
- Dictionary<TKey><TValue>.Keys
- Dictionary<TKey><TValue>.Values

System.Collections.Generic.Dictionary<TKey><TValue>.Enumerator structure

[Dictionary<TKey><TValue>.Enumerator Methods]
Dictionary<TKey><TValue>.Enumerator.Dispose
Dictionary<TKey><TValue>.Enumerator.MoveNext
[Dictionary<TKey><TValue>.Enumerator Properties]
Dictionary<TKey><TValue>.Enumerator.Current

System.Collections.Generic.Dictionary<TKey><TValue>.KeyCollection class

Dictionary<TKey><TValue>.KeyCollection.Dictionary<TKey><TValue>.KeyCollection
[Dictionary<TKey><TValue>.KeyCollection Methods]
Dictionary<TKey><TValue>.KeyCollection.CopyTo
Dictionary<TKey><TValue>.KeyCollection.GetEnumerator
[Dictionary<TKey><TValue>.KeyCollection Properties]
Dictionary<TKey><TValue>.KeyCollection.Count

System.Collections.Generic.Dictionary<TKey><TValue>.KeyCollection.Enumerator structure

[Dictionary<TKey><TValue>.KeyCollection.Enumerator Methods]
Dictionary<TKey><TValue>.KeyCollection.Enumerator.Dispose
Dictionary<TKey><TValue>.KeyCollection.Enumerator.MoveNext
[Dictionary<TKey><TValue>.KeyCollection.Enumerator Properties]
Dictionary<TKey><TValue>.KeyCollection.Enumerator.Current

System.Collections.Generic.Dictionary<TKey><TValue>.ValueCollection class

Dictionary<TKey><TValue>.ValueCollection.Dictionary<TKey><TValue>.ValueCollection
[Dictionary<TKey><TValue>.ValueCollection Methods]
Dictionary<TKey><TValue>.ValueCollection.CopyTo
Dictionary<TKey><TValue>.ValueCollection.GetEnumerator
[Dictionary<TKey><TValue>.ValueCollection Properties]
Dictionary<TKey><TValue>.ValueCollection.Count

System.Collections.Generic.Dictionary<TKey><TValue>.ValueCollection.Enumerator structure

[Dictionary<TKey><TValue>.ValueCollection.Enumerator Methods]
Dictionary<TKey><TValue>.ValueCollection.Enumerator.Dispose
Dictionary<TKey><TValue>.ValueCollection.Enumerator.MoveNext
[Dictionary<TKey><TValue>.ValueCollection.Enumerator Properties]
Dictionary<TKey><TValue>.ValueCollection.Enumerator.Current

System.Collections.Generic.EqualityComparer<T> class

EqualityComparer<T>.EqualityComparer<T>
[EqualityComparer<T> Methods]
EqualityComparer<T>.Equals
EqualityComparer<T>.GetHashCode
[EqualityComparer<T> Properties]
EqualityComparer<T>.Default

System.Collections.Generic.ICollection<T> interface

[ICollection<T> Methods]
ICollection<T>.Add
ICollection<T>.Clear
ICollection<T>.Contains
ICollection<T>.CopyTo
ICollection<T>.Remove
[ICollection<T> Properties]
ICollection<T>.Count
ICollection<T>.IsReadOnly

System.Collections.Generic.IComparer<T> interface

[IComparer<T> Methods]
IComparer<T>.Compare

System.Collections.Generic.IDictionary<TKey><TValue> interface

[IDictionary<TKey><TValue> Methods]

IDictionary<TKey> <TValue>.Add
IDictionary<TKey> <TValue>.ContainsKey
IDictionary<TKey> <TValue>.Remove
IDictionary<TKey> <TValue>.TryGetValue
[IDictionary<TKey> <TValue> Properties]
IDictionary<TKey> <TValue>.Item
IDictionary<TKey> <TValue>.Keys
IDictionary<TKey> <TValue>.Values

System.Collections.Generic.IEnumerable<T> interface

[IEnumerable<T> Methods]
IEnumerable<T>.GetEnumerator

System.Collections.Generic.IEnumerator<T> interface

[IEnumerator<T> Properties]
IEnumerator<T>.Current

System.Collections.Generic.IEqualityComparer<T> interface

[IEqualityComparer<T> Methods]
IEqualityComparer<T>.Equals
IEqualityComparer<T>.GetHashCode

System.Collections.Generic.IList<T> interface

[IList<T> Methods]
IList<T>.IndexOf
IList<T>.Insert
IList<T>.RemoveAt
[IList<T> Properties]
IList<T>.Item

System.Collections.Generic.KeyNotFoundException class

KeyNotFoundException.KeyNotFoundException

System.Collections.Generic.KeyValuePair<TKey> <TValue> structure

KeyValuePair<TKey> <TValue>.KeyValuePair<TKey> <TValue>
[KeyValuePair<TKey> <TValue> Methods]
KeyValuePair<TKey> <TValue>.ToString
[KeyValuePair<TKey> <TValue> Properties]
KeyValuePair<TKey> <TValue>.Key
KeyValuePair<TKey> <TValue>.Value

System.Collections.Generic.LinkedList<T> class

LinkedList<T>.LinkedList<T>
[LinkedList<T> Methods]
LinkedList<T>.AddAfter
LinkedList<T>.AddBefore
LinkedList<T>.AddFirst
LinkedList<T>.AddLast
LinkedList<T>.Clear
LinkedList<T>.Contains
LinkedList<T>.CopyTo
LinkedList<T>.Find
LinkedList<T>.FindLast
LinkedList<T>.GetEnumerator
LinkedList<T>.Remove
LinkedList<T>.RemoveFirst
LinkedList<T>.RemoveLast
[LinkedList<T> Properties]
LinkedList<T>.Count
LinkedList<T>.First
LinkedList<T>.Last

System.Collections.Generic.LinkedList<T>.Enumerator structure

[LinkedList<T>.Enumerator Methods]
LinkedList<T>.Enumerator.Dispose
LinkedList<T>.Enumerator.MoveNext
[LinkedList<T>.Enumerator Properties]
LinkedList<T>.Enumerator.Current

System.Collections.Generic.LinkedListNode<T> class

LinkedListNode<T>.LinkedListNode<T>
[LinkedListNode<T> Properties]
LinkedListNode<T>.List
LinkedListNode<T>.Next
LinkedListNode<T>.Previous
LinkedListNode<T>.Value

System.Collections.Generic.List<T> class

List<T>.List<T>
[List<T> Methods]
List<T>.Add
List<T>.AddRange
List<T>.AsReadOnly
List<T>.BinarySearch
List<T>.Clear
List<T>.Contains
List<T>.ConvertAll<TOutput>
List<T>.CopyTo
List<T>.Exists
List<T>.Find
List<T>.FindAll
List<T>.FindIndex
List<T>.FindLast
List<T>.FindLastIndex
List<T>.ForEach
List<T>.GetEnumerator
List<T>.GetRange
List<T>.IndexOf
List<T>.Insert
List<T>.InsertRange
List<T>.LastIndexOf
List<T>.Remove
List<T>.RemoveAll
List<T>.RemoveAt
List<T>.RemoveRange
List<T>.Reverse
List<T>.Sort
List<T>.ToArray
List<T>.TrimExcess
List<T>.TrueForAll
[List<T> Properties]
List<T>.Capacity
List<T>.Count
List<T>.Item

System.Collections.Generic.List<T>.Enumerator structure

[List<T>.Enumerator Methods]
List<T>.Enumerator.Dispose
List<T>.Enumerator.MoveNext
[List<T>.Enumerator Properties]
List<T>.Enumerator.Current

System.Collections.Generic.Queue<T> class

Queue<T>.Queue<T>

[Queue<T> Methods]
Queue<T>.Clear
Queue<T>.Contains
Queue<T>.CopyTo
Queue<T>.Dequeue
Queue<T>.Enqueue
Queue<T>.GetEnumerator
Queue<T>.Peek
Queue<T>.ToArray
Queue<T>.TrimExcess
[Queue<T> Properties]
Queue<T>.Count

System.Collections.Generic.Queue<T>.Enumerator structure

[Queue<T>.Enumerator Methods]
Queue<T>.Enumerator.Dispose
Queue<T>.Enumerator.MoveNext
[Queue<T>.Enumerator Properties]
Queue<T>.Enumerator.Current

System.Collections.Generic.SortedList<TKey><TValue> class

SortedList<TKey><TValue>.SortedList<TKey><TValue>
[SortedList<TKey><TValue> Methods]
SortedList<TKey><TValue>.Add
SortedList<TKey><TValue>.Clear
SortedList<TKey><TValue>.ContainsKey
SortedList<TKey><TValue>.ContainsValue
SortedList<TKey><TValue>.GetEnumerator
SortedList<TKey><TValue>.IndexOfKey
SortedList<TKey><TValue>.IndexOfValue
SortedList<TKey><TValue>.Remove
SortedList<TKey><TValue>.RemoveAt
SortedList<TKey><TValue>.TrimExcess
SortedList<TKey><TValue>.TryGetValue
[SortedList<TKey><TValue> Properties]
SortedList<TKey><TValue>.Capacity
SortedList<TKey><TValue>.Comparer
SortedList<TKey><TValue>.Count
SortedList<TKey><TValue>.Item
SortedList<TKey><TValue>.Keys
SortedList<TKey><TValue>.Values

System.Collections.Generic.Stack<T> class

Stack<T>.Stack<T>
[Stack<T> Methods]
Stack<T>.Clear
Stack<T>.Contains
Stack<T>.CopyTo
Stack<T>.GetEnumerator
Stack<T>.Peek
Stack<T>.Pop
Stack<T>.Push
Stack<T>.ToArray
Stack<T>.TrimExcess
[Stack<T> Properties]
Stack<T>.Count

System.Collections.Generic.Stack<T>.Enumerator structure

[Stack<T>.Enumerator Methods]
Stack<T>.Enumerator.Dispose
Stack<T>.Enumerator.MoveNext
[Stack<T>.Enumerator Properties]

Stack<T>.Enumerator.Current

System.Collections.ObjectModel namespace

System.Collections.ObjectModel.Collection<T> class

Collection<T>.Collection<T>
[Collection<T> Methods]
Collection<T>.Add
Collection<T>.Clear
Collection<T>.ClearItems
Collection<T>.Contains
Collection<T>.CopyTo
Collection<T>.GetEnumerator
Collection<T>.IndexOf
Collection<T>.Insert
Collection<T>.InsertItem
Collection<T>.Remove
Collection<T>.RemoveAt
Collection<T>.RemoveItem
Collection<T>.SetItem
[Collection<T> Properties]
Collection<T>.Count
Collection<T>.Item
Collection<T>.Items

System.Collections.ObjectModel.KeyedCollection<TKey><TItem> class

KeyedCollection<TKey><TItem>.KeyedCollection<TKey><TItem>
[KeyedCollection<TKey><TItem> Methods]
KeyedCollection<TKey><TItem>.ChangeItemKey
KeyedCollection<TKey><TItem>.ClearItems
KeyedCollection<TKey><TItem>.Contains
KeyedCollection<TKey><TItem>.GetKeyForItem
KeyedCollection<TKey><TItem>.InsertItem
KeyedCollection<TKey><TItem>.Remove
KeyedCollection<TKey><TItem>.RemoveItem
KeyedCollection<TKey><TItem>.SetItem
[KeyedCollection<TKey><TItem> Properties]
KeyedCollection<TKey><TItem>.Comparer
KeyedCollection<TKey><TItem>.Dictionary
KeyedCollection<TKey><TItem>.Item

System.Collections.ObjectModel.ReadOnlyCollection<T> class

ReadOnlyCollection<T>.ReadOnlyCollection<T>
[ReadOnlyCollection<T> Methods]
ReadOnlyCollection<T>.Contains
ReadOnlyCollection<T>.CopyTo
ReadOnlyCollection<T>.GetEnumerator
ReadOnlyCollection<T>.IndexOf
[ReadOnlyCollection<T> Properties]
ReadOnlyCollection<T>.Count
ReadOnlyCollection<T>.Item
ReadOnlyCollection<T>.Items

System.Collections.Specialized namespace

System.Collections.Specialized.BitVector32 structure

BitVector32.BitVector32
[BitVector32 Methods]
BitVector32.CreateMask
BitVector32.CreateSection

- BitVector32.Equals
- BitVector32.GetHashCode
- BitVector32.ToString
- [BitVector32 Properties]
- BitVector32.Data
- BitVector32.Item
- BitVector32.Item

System.Collections.Specialized.BitVector32.Section structure

- [BitVector32.Section Methods]
- BitVector32.Section.Equals
- BitVector32.Section.GetHashCode
- BitVector32.Section.op_Equality
- BitVector32.Section.op_Inequality
- BitVector32.Section.ToString
- [BitVector32.Section Properties]
- BitVector32.Section.Mask
- BitVector32.Section.Offset

System.Collections.Specialized.HybridDictionary class

- HybridDictionary.HybridDictionary
- [HybridDictionary Methods]
- HybridDictionary.Add
- HybridDictionary.Clear
- HybridDictionary.Contains
- HybridDictionary.CopyTo
- HybridDictionary.GetEnumerator
- HybridDictionary.Remove
- [HybridDictionary Properties]
- HybridDictionary.Count
- HybridDictionary.IsFixedSize
- HybridDictionary.IsReadOnly
- HybridDictionary.IsSynchronized
- HybridDictionary.Item
- HybridDictionary.Keys
- HybridDictionary.SyncRoot
- HybridDictionary.Values

System.Collections.Specialized.ListDictionary class

- ListDictionary.ListDictionary
- [ListDictionary Methods]
- ListDictionary.Add
- ListDictionary.Clear
- ListDictionary.Contains
- ListDictionary.CopyTo
- ListDictionary.GetEnumerator
- ListDictionary.Remove
- [ListDictionary Properties]
- ListDictionary.Count
- ListDictionary.IsFixedSize
- ListDictionary.IsReadOnly
- ListDictionary.IsSynchronized
- ListDictionary.Item
- ListDictionary.Keys
- ListDictionary.SyncRoot
- ListDictionary.Values

System.Collections.Specialized.NameObjectCollectionBase class

- NameObjectCollectionBase.NameObjectCollectionBase
- [NameObjectCollectionBase Methods]
- NameObjectCollectionBase.BaseAdd
- NameObjectCollectionBase.BaseClear

- NameObjectCollectionBase.BaseGet
- NameObjectCollectionBase.BaseGetAllKeys
- NameObjectCollectionBase.BaseGetKey
- NameObjectCollectionBase.BaseHasKeys
- NameObjectCollectionBase.BaseRemove
- NameObjectCollectionBase.BaseRemoveAt
- NameObjectCollectionBase.BaseSet
- NameObjectCollectionBase.GetEnumerator
- [NameObjectCollectionBase Properties]
- NameObjectCollectionBase.Count
- NameObjectCollectionBase.IsReadOnly
- NameObjectCollectionBase.Keys

System.Collections.Specialized.NameObjectCollectionBase.KeysCollection class

- [NameObjectCollectionBase.KeysCollection Methods]
- NameObjectCollectionBase.KeysCollection.Get
- NameObjectCollectionBase.KeysCollection.GetEnumerator
- [NameObjectCollectionBase.KeysCollection Properties]
- NameObjectCollectionBase.KeysCollection.Count
- NameObjectCollectionBase.KeysCollection.Item

System.Collections.Specialized.NameValueCollection class

- NameValueCollection.NameValueCollection
- [NameValueCollection Methods]
- NameValueCollection.Add
- NameValueCollection.Clear
- NameValueCollection.Get
- NameValueCollection.GetKey
- NameValueCollection.GetValues
- NameValueCollection.HasKeys
- NameValueCollection.InvalidateCachedArrays
- NameValueCollection.Remove
- NameValueCollection.Set
- [NameValueCollection Properties]
- NameValueCollection.AllKeys
- NameValueCollection.Item
- NameValueCollection.Item

System.Collections.Specialized.StringDictionary class

- StringDictionary.StringDictionary
- [StringDictionary Methods]
- StringDictionary.Add
- StringDictionary.Clear
- StringDictionary.ContainsKey
- StringDictionary.ContainsValue
- StringDictionary.CopyTo
- StringDictionary.GetEnumerator
- StringDictionary.Remove
- [StringDictionary Properties]
- StringDictionary.Count
- StringDictionary.IsSynchronized
- StringDictionary.Item
- StringDictionary.Keys
- StringDictionary.SyncRoot
- StringDictionary.Values

System.ComponentModel namespace

System.ComponentModel.AddingNewEventArgs class

- AddingNewEventArgs.AddingNewEventArgs
- [AddingNewEventArgs Properties]

AddingNewEventArgs.NewObject

System.ComponentModel.AddingNewEventHandler delegate

System.ComponentModel.AsyncCompletedEventArgs class

AsyncCompletedEventArgs.AsyncCompletedEventArgs
[AsyncCompletedEventArgs Methods]
AsyncCompletedEventArgs.RaiseExceptionIfNecessary
[AsyncCompletedEventArgs Properties]
AsyncCompletedEventArgs.Cancelled
AsyncCompletedEventArgs.Error
AsyncCompletedEventArgs.UserState

System.ComponentModel.AsyncCompletedEventHandler delegate

System.ComponentModel.AttributeCollection class

AttributeCollection.AttributeCollection
[AttributeCollection Fields]
AttributeCollection.Empty
[AttributeCollection Methods]
AttributeCollection.Contains
AttributeCollection.CopyTo
AttributeCollection.GetDefaultAttribute
AttributeCollection.GetEnumerator
AttributeCollection.Matches
[AttributeCollection Properties]
AttributeCollection.Count
AttributeCollection.Item
AttributeCollection.Item

System.ComponentModel.BindingList<T> class

BindingList<T>.BindingList<T>
[BindingList<T> Methods]
BindingList<T>.AddNew
BindingList<T>.AddNewCore
BindingList<T>.ApplySortCore
BindingList<T>.CancelNew
BindingList<T>.ClearItems
BindingList<T>.EndNew
BindingList<T>.FindCore
BindingList<T>.InsertItem
BindingList<T>.OnAddingNew
BindingList<T>.OnListChanged
BindingList<T>.RemoveItem
BindingList<T>.RemoveSortCore
BindingList<T>.ResetBindings
BindingList<T>.ResetItem
BindingList<T>.SetItem
[BindingList<T> Properties]
BindingList<T>.AllowEdit
BindingList<T>.AllowNew
BindingList<T>.AllowRemove
BindingList<T>.IsSortedCore
BindingList<T>.RaiseListChangedEvents
BindingList<T>.SortDirectionCore
BindingList<T>.SortPropertyCore
BindingList<T>.SupportsChangeNotificationCore
BindingList<T>.SupportsSearchingCore
BindingList<T>.SupportsSortingCore
[BindingList<T> Events]
BindingList<T>.AddingNew
BindingList<T>.ListChanged

System.ComponentModel.CancelEventArgs class

CancelEventArgs.CancelEventArgs
[CancelEventArgs Properties]
CancelEventArgs.Cancel

System.ComponentModel.CancelEventHandler delegate

System.ComponentModel.CollectionChangeAction enumeration

System.ComponentModel.CollectionChangeEventArgs class

CollectionChangeEventArgs.CollectionChangeEventArgs
[CollectionChangeEventArgs Properties]
CollectionChangeEventArgs.Action
CollectionChangeEventArgs.Element

System.ComponentModel.CollectionChangeEventHandler delegate

System.ComponentModel.Component class

Component.Component
[Component Methods]
Component.Dispose
Component.Finalize
[Component Properties]
Component.Events
Component.Site
[Component Events]
Component.Disposed

System.ComponentModel.ComponentResourceManager class

ComponentResourceManager.ComponentResourceManager
[ComponentResourceManager Methods]
ComponentResourceManager.ApplyResources

System.ComponentModel.Container class

Container.Container
[Container Methods]
Container.Add
Container.CreateSite
Container.Dispose
Container.GetService
Container.Remove

System.ComponentModel.DataObjectAttribute class

DataObjectAttribute.DataObjectAttribute
[DataObjectAttribute Fields]
DataObjectAttribute.DataObject
DataObjectAttribute.Default
DataObjectAttribute.NonDataObject
[DataObjectAttribute Methods]
DataObjectAttribute.Equals
DataObjectAttribute.GetHashCode
[DataObjectAttribute Properties]
DataObjectAttribute.IsDataObject

System.ComponentModel.DataObjectMethodAttribute class

DataObjectMethodAttribute.DataObjectMethodAttribute
[DataObjectMethodAttribute Methods]
DataObjectMethodAttribute.Equals
DataObjectMethodAttribute.GetHashCode
DataObjectMethodAttribute.Match
[DataObjectMethodAttribute Properties]

DataObjectMethodAttribute.IsDefault
DataObjectMethodAttribute.MethodType

System.ComponentModel.DataObjectMethodType enumeration

System.ComponentModel.DefaultValueAttribute class

DefaultValueAttribute.DefaultValueAttribute
[DefaultValueAttribute Methods]
DefaultValueAttribute.Equals
DefaultValueAttribute.GetHashCode
[DefaultValueAttribute Properties]
DefaultValueAttribute.Value

System.ComponentModel.DesignerCategoryAttribute class

DesignerCategoryAttribute.DesignerCategoryAttribute
[DesignerCategoryAttribute Fields]
DesignerCategoryAttribute.Component
DesignerCategoryAttribute.Default
DesignerCategoryAttribute.Form
DesignerCategoryAttribute.Generic
[DesignerCategoryAttribute Methods]
DesignerCategoryAttribute.Equals
DesignerCategoryAttribute.GetHashCode
[DesignerCategoryAttribute Properties]
DesignerCategoryAttribute.Category

System.ComponentModel.DesignTimeVisibleAttribute class

DesignTimeVisibleAttribute.DesignTimeVisibleAttribute
[DesignTimeVisibleAttribute Fields]
DesignTimeVisibleAttribute.Default
DesignTimeVisibleAttribute.No
DesignTimeVisibleAttribute.Yes
[DesignTimeVisibleAttribute Methods]
DesignTimeVisibleAttribute.Equals
DesignTimeVisibleAttribute.GetHashCode
[DesignTimeVisibleAttribute Properties]
DesignTimeVisibleAttribute.Visible

System.ComponentModel.EditorBrowsableAttribute class

EditorBrowsableAttribute.EditorBrowsableAttribute
[EditorBrowsableAttribute Methods]
EditorBrowsableAttribute.Equals
EditorBrowsableAttribute.GetHashCode
[EditorBrowsableAttribute Properties]
EditorBrowsableAttribute.State

System.ComponentModel.EditorBrowsableState enumeration

System.ComponentModel.EventDescriptor class

EventDescriptor.EventDescriptor
[EventDescriptor Methods]
EventDescriptor.AddEventHandler
EventDescriptor.RemoveEventHandler
[EventDescriptor Properties]
EventDescriptor.ComponentType
EventDescriptor.EventType
EventDescriptor.IsMulticast

System.ComponentModel.EventDescriptorCollection class

EventDescriptorCollection.EventDescriptorCollection
[EventDescriptorCollection Fields]
EventDescriptorCollection.Empty

[EventDescriptorCollection Methods]
EventDescriptorCollection.Add
EventDescriptorCollection.Clear
EventDescriptorCollection.Contains
EventDescriptorCollection.Find
EventDescriptorCollection.GetEnumerator
EventDescriptorCollection.IndexOf
EventDescriptorCollection.Insert
EventDescriptorCollection.Remove
EventDescriptorCollection.RemoveAt
[EventDescriptorCollection Properties]
EventDescriptorCollection.Count
EventDescriptorCollection.Item
EventDescriptorCollection.Item

System.ComponentModel.EventHandlerList class

EventHandlerList.EventHandlerList
[EventHandlerList Methods]
EventHandlerList.AddHandler
EventHandlerList.AddHandlers
EventHandlerList.Dispose
EventHandlerList.RemoveHandler
[EventHandlerList Properties]
EventHandlerList.Item

System.ComponentModel.IBindingList interface

[IBindingList Methods]
IBindingList.AddIndex
IBindingList.AddNew
IBindingList.ApplySort
IBindingList.Find
IBindingList.RemoveIndex
IBindingList.RemoveSort
[IBindingList Properties]
IBindingList.AllowEdit
IBindingList.AllowNew
IBindingList.AllowRemove
IBindingList.IsSorted
IBindingList.SortDirection
IBindingList.SortProperty
IBindingList.SupportsChangeNotification
IBindingList.SupportsSearching
IBindingList.SupportsSorting
[IBindingList Events]
IBindingList.ListChanged

System.ComponentModel.IBindingListView interface

[IBindingListView Methods]
IBindingListView.ApplySort
IBindingListView.RemoveFilter
[IBindingListView Properties]
IBindingListView.Filter
IBindingListView.SortDescriptions
IBindingListView.SupportsAdvancedSorting
IBindingListView.SupportsFiltering

System.ComponentModel.ICancelAddNew interface

[ICancelAddNew Methods]
ICancelAddNew.CancelNew
ICancelAddNew.EndNew

System.ComponentModel.IChangeTracking interface

[IChangeTracking Methods]
IChangeTracking.AcceptChanges
[IChangeTracking Properties]
IChangeTracking.IsChanged

System.ComponentModel.IComponent interface

[IComponent Properties]
IComponent.Site
[IComponent Events]
IComponent.Disposed

System.ComponentModel.IContainer interface

[IContainer Methods]
IContainer.Add
IContainer.Remove

System.ComponentModel.ICustomTypeDescriptor interface

[ICustomTypeDescriptor Methods]
ICustomTypeDescriptor.GetAttributes
ICustomTypeDescriptor.GetClassName
ICustomTypeDescriptor.GetComponentName
ICustomTypeDescriptor.GetConverter
ICustomTypeDescriptor.GetDefaultEvent
ICustomTypeDescriptor.GetDefaultProperty
ICustomTypeDescriptor.GetEditor
ICustomTypeDescriptor.GetEvents
ICustomTypeDescriptor.GetProperties
ICustomTypeDescriptor.GetPropertyOwner

System.ComponentModel.IDataErrorInfo interface

[IDataErrorInfo Properties]
IDataErrorInfo.Error
IDataErrorInfo.Item

System.ComponentModel.IEditableObject interface

[IEditableObject Methods]
IEditableObject.BeginEdit
IEditableObject.CancelEdit
IEditableObject.EndEdit

System.ComponentModel.IListSource interface

[IListSource Methods]
IListSource.GetList
[IListSource Properties]
IListSource.ContainsListCollection

System.ComponentModel.INotifyPropertyChanged interface

[INotifyPropertyChanged Events]
INotifyPropertyChanged.PropertyChanged

System.ComponentModel.IRaiseItemChangedEvents interface

[IRaiseItemChangedEvents Properties]
IRaiseItemChangedEvents.RaiseItemChangedEvents

System.ComponentModel.IReversibleChangeTracking interface

[IReversibleChangeTracking Methods]
IReversibleChangeTracking.RejectChanges

System.ComponentModel.ISite interface

[ISite Properties]
ISite.Component
ISite.Container

ISite.DesignMode
ISite.Name

System.ComponentModel.ISupportInitialize interface

[ISupportInitialize Methods]
ISupportInitialize.BeginInit
ISupportInitialize.EndInit

System.ComponentModel.ISupportInitializeNotification interface

[ISupportInitializeNotification Properties]
ISupportInitializeNotification.IsInitialized
[ISupportInitializeNotification Events]
ISupportInitializeNotification.Initialized

System.ComponentModel.ITypedList interface

[ITypedList Methods]
ITypedList.GetItemProperties
ITypedList.GetListName

System.ComponentModel.ListChangedEventArgs class

ListChangedEventArgs.ListChangedEventArgs
[ListChangedEventArgs Properties]
ListChangedEventArgs.ListChangedType
ListChangedEventArgs.NewIndex
ListChangedEventArgs.OldIndex

System.ComponentModel.ListChangedEventHandler delegate

System.ComponentModel.ListChangedType enumeration

System.ComponentModel.ListSortDescription class

ListSortDescription.ListSortDescription
[ListSortDescription Properties]
ListSortDescription.PropertyDescriptor
ListSortDescription.SortDirection

System.ComponentModel.ListSortDescriptionCollection class

ListSortDescriptionCollection.ListSortDescriptionCollection
[ListSortDescriptionCollection Methods]
ListSortDescriptionCollection.Contains
ListSortDescriptionCollection.CopyTo
ListSortDescriptionCollection.IndexOf
[ListSortDescriptionCollection Properties]
ListSortDescriptionCollection.Count
ListSortDescriptionCollection.Item

System.ComponentModel.ListSortDirection enumeration

System.ComponentModel.MarshalByValueComponent class

MarshalByValueComponent.MarshalByValueComponent
[MarshalByValueComponent Methods]
MarshalByValueComponent.Dispose
MarshalByValueComponent.Finalize
MarshalByValueComponent.GetService
MarshalByValueComponent.ToString
[MarshalByValueComponent Properties]
MarshalByValueComponent.Container
MarshalByValueComponent.DesignMode
MarshalByValueComponent.Events
MarshalByValueComponent.Site
[MarshalByValueComponent Events]
MarshalByValueComponent.Disposed

System.ComponentModel.MemberDescriptor class

MemberDescriptor.MemberDescriptor
[MemberDescriptor Methods]
MemberDescriptor.CreateAttributeCollection
MemberDescriptor.FillAttributes
MemberDescriptor.FindMethod
MemberDescriptor.GetHashCode
[MemberDescriptor Properties]
MemberDescriptor.AttributeArray
MemberDescriptor.DisplayName
MemberDescriptor.Name
MemberDescriptor.NameHashCode

System.ComponentModel.PropertyChangedEventArgs class

PropertyChangedEventArgs.PropertyChangedEventArgs
[PropertyChangedEventArgs Properties]
PropertyChangedEventArgs.PropertyName

System.ComponentModel.PropertyChangedEventHandler delegate

System.ComponentModel.PropertyDescriptor class

PropertyDescriptor.PropertyDescriptor
[PropertyDescriptor Methods]
PropertyDescriptor.AddValueChanged
PropertyDescriptor.CanResetValue
PropertyDescriptor.GetValue
PropertyDescriptor.GetValueChangedHandler
PropertyDescriptor.OnValueChanged
PropertyDescriptor.RemoveValueChanged
PropertyDescriptor.ResetValue
PropertyDescriptor.SetValue
PropertyDescriptor.ShouldSerializeValue
[PropertyDescriptor Properties]
PropertyDescriptor.ComponentType
PropertyDescriptor.IsReadOnly
PropertyDescriptor.PropertyType
PropertyDescriptor.SupportsChangeEvents

System.ComponentModel.PropertyDescriptorCollection class

PropertyDescriptorCollection.PropertyDescriptorCollection
[PropertyDescriptorCollection Fields]
PropertyDescriptorCollection.Empty
[PropertyDescriptorCollection Methods]
PropertyDescriptorCollection.Add
PropertyDescriptorCollection.Clear
PropertyDescriptorCollection.Contains
PropertyDescriptorCollection.CopyTo
PropertyDescriptorCollection.Find
PropertyDescriptorCollection.GetEnumerator
PropertyDescriptorCollection.IndexOf
PropertyDescriptorCollection.Insert
PropertyDescriptorCollection.Remove
PropertyDescriptorCollection.RemoveAt
[PropertyDescriptorCollection Properties]
PropertyDescriptorCollection.Count
PropertyDescriptorCollection.Item
PropertyDescriptorCollection.Item

System.ComponentModel.TypeConverter class

TypeConverter.TypeConverter

System.ComponentModel.TypeDescriptor class

[TypeDescriptor Methods]

TypeDescriptor.GetEvents

TypeDescriptor.GetProperties

System.ComponentModel.Win32Exception class

Win32Exception.Win32Exception

[Win32Exception Properties]

Win32Exception.NativeErrorCode

System.Configuration.Assemblies namespace

System.Configuration.Assemblies.AssemblyHashAlgorithm enumeration

System.Configuration.Assemblies.AssemblyVersionCompatibility enumeration

System.Diagnostics namespace

System.Diagnostics.ConditionalAttribute class

ConditionalAttribute.ConditionalAttribute

[ConditionalAttribute Properties]

ConditionalAttribute.ConditionString

System.Diagnostics.Debug class

[Debug Methods]

Debug.Assert

Debug.Close

Debug.Fail

Debug.Flush

Debug.Write

Debug.Writelf

Debug.WriteLine

Debug.WriteLinef

[Debug Properties]

Debug.AutoFlush

Debug.IndentLevel

Debug.IndentSize

Debug.Listeners

System.Diagnostics.DebuggableAttribute class

DebuggableAttribute.DebuggableAttribute

System.Diagnostics.DebuggableAttribute.DebuggingModes enumeration

System.Diagnostics.Debugger class

Debugger.Debugger

[Debugger Methods]

Debugger.Break

Debugger.Launch

[Debugger Properties]

Debugger.IsAttached

System.Diagnostics.DebuggerHiddenAttribute class

DebuggerHiddenAttribute.DebuggerHiddenAttribute

System.Diagnostics.DebuggerNonUserCodeAttribute class

DebuggerNonUserCodeAttribute.DebuggerNonUserCodeAttribute

System.Diagnostics.DebuggerStepThroughAttribute class

DebuggerStepThroughAttribute.DebuggerStepThroughAttribute

System.Diagnostics.DefaultTraceListener class

DefaultTraceListener.DefaultTraceListener

[DefaultTraceListener Methods]

DefaultTraceListener.Fail

DefaultTraceListener.Write

DefaultTraceListener.WriteLine

System.Diagnostics.Stopwatch class

Stopwatch.Stopwatch

[Stopwatch Fields]

Stopwatch.Frequency

Stopwatch.IsHighResolution

[Stopwatch Methods]

Stopwatch.GetTimestamp

Stopwatch.Reset

Stopwatch.Start

Stopwatch.StartNew

Stopwatch.Stop

[Stopwatch Properties]

Stopwatch.Elapsed

Stopwatch.ElapsedMilliseconds

Stopwatch.ElapsedTicks

Stopwatch.IsRunning

System.Diagnostics.TextWriterTraceListener class

TextWriterTraceListener.TextWriterTraceListener

[TextWriterTraceListener Methods]

TextWriterTraceListener.Close

TextWriterTraceListener.Dispose

TextWriterTraceListener.Flush

TextWriterTraceListener.Write

TextWriterTraceListener.WriteLine

[TextWriterTraceListener Properties]

TextWriterTraceListener.Writer

System.Diagnostics.Trace class

[Trace Methods]

Trace.Assert

Trace.Close

Trace.Fail

Trace.Flush

Trace.Write

Trace.Writelf

Trace.WriteLine

Trace.WriteLineelf

[Trace Properties]

Trace.AutoFlush

Trace.IndentLevel

Trace.IndentSize

Trace.Listeners

System.Diagnostics.TraceListener class

TraceListener.TraceListener

[TraceListener Methods]

TraceListener.Close

TraceListener.Dispose

TraceListener.Fail

TraceListener.Flush

TraceListener.Write

TraceListener.WriteIndent

TraceListener.WriteLine

[TraceListener Properties]
TraceListener.IndentLevel
TraceListener.IndentSize
TraceListener.Name
TraceListener.NeedIndent

System.Diagnostics.TraceListenerCollection class

[TraceListenerCollection Methods]
TraceListenerCollection.Add
TraceListenerCollection.Clear
TraceListenerCollection.Contains
TraceListenerCollection.CopyTo
TraceListenerCollection.GetEnumerator
TraceListenerCollection.IndexOf
TraceListenerCollection.Insert
TraceListenerCollection.Remove
TraceListenerCollection.RemoveAt
[TraceListenerCollection Properties]
TraceListenerCollection.Count
TraceListenerCollection.Item
TraceListenerCollection.Item

System.Diagnostics.CodeAnalysis namespace

System.Diagnostics.CodeAnalysis.SuppressMessageAttribute class

SuppressMessageAttribute.SuppressMessageAttribute
[SuppressMessageAttribute Properties]
SuppressMessageAttribute.Category
SuppressMessageAttribute.CheckId
SuppressMessageAttribute.Justification
SuppressMessageAttribute.MessageId
SuppressMessageAttribute.Scope
SuppressMessageAttribute.Target

System.Globalization namespace

System.Globalization.Calendar class

Calendar.Calendar
[Calendar Fields]
Calendar.CurrentEra
[Calendar Methods]
Calendar.AddDays
Calendar.AddHours
Calendar.AddMilliseconds
Calendar.AddMinutes
Calendar.AddMonths
Calendar.AddSeconds
Calendar.AddWeeks
Calendar.AddYears
Calendar.GetDayOfMonth
Calendar.GetDayOfWeek
Calendar.GetDayOfYear
Calendar.GetDaysInMonth
Calendar.GetDaysInYear
Calendar.GetEra
Calendar.GetHour
Calendar.GetMilliseconds
Calendar.GetMinute
Calendar.GetMonth
Calendar.GetMonthsInYear

- Calendar.GetSecond
- Calendar.GetWeekOfYear
- Calendar.GetYear
- Calendar.IsLeapDay
- Calendar.IsLeapMonth
- Calendar.IsLeapYear
- Calendar.ToDateTime
- Calendar.ToFourDigitYear

[Calendar Properties]

- Calendar.Eras
- Calendar.IsReadOnly
- Calendar.MaxSupportedDateTime
- Calendar.MinSupportedDateTime
- Calendar.TwoDigitYearMax

System.Globalization.CalendarWeekRule enumeration

System.Globalization.CharUnicodeInfo class

[CharUnicodeInfo Methods]

- CharUnicodeInfo.GetNumericValue
- CharUnicodeInfo.GetUnicodeCategory

System.Globalization.CompareInfo class

[CompareInfo Methods]

- CompareInfo.Compare
- CompareInfo.Equals
- CompareInfo.GetCompareInfo
- CompareInfo.GetHashCode
- CompareInfo.IndexOf
- CompareInfo.IsPrefix
- CompareInfo.IsSuffix
- CompareInfo.LastIndexOf
- CompareInfo.ToString

[CompareInfo Properties]

- CompareInfo.LCID
- CompareInfo.Name

System.Globalization.CompareOptions enumeration

System.Globalization.CultureInfo class

CultureInfo.CultureInfo

[CultureInfo Methods]

- CultureInfo.ClearCachedData
- CultureInfo.Clone
- CultureInfo.CreateSpecificCulture
- CultureInfo.Equals
- CultureInfo.GetCultures
- CultureInfo.GetFormat
- CultureInfo.GetHashCode
- CultureInfo.ReadOnly
- CultureInfo.ToString

[CultureInfo Properties]

- CultureInfo.Calendar
- CultureInfo.CompareInfo
- CultureInfo.CurrentCulture
- CultureInfo.CurrentUICulture
- CultureInfo.DateTimeFormat
- CultureInfo.EnglishName
- CultureInfo.InvariantCulture
- CultureInfo.IsNeutralCulture
- CultureInfo.IsReadOnly
- CultureInfo.LCID

CultureInfo.Name
CultureInfo.NativeName
CultureInfo.NumberFormat
CultureInfo.OptionalCalendars
CultureInfo.Parent
CultureInfo.TextInfo
CultureInfo.ThreeLetterISOLanguageName
CultureInfo.ThreeLetterWindowsLanguageName
CultureInfo.TwoLetterISOLanguageName
CultureInfo.UseUserOverride

System.Globalization.CultureTypes enumeration

System.Globalization.DateTimeFormatInfo class

DateTimeFormatInfo.DateTimeFormatInfo
[DateTimeFormatInfo Methods]
DateTimeFormatInfo.Clone
DateTimeFormatInfo.GetAbbreviatedDayName
DateTimeFormatInfo.GetAbbreviatedEraName
DateTimeFormatInfo.GetAbbreviatedMonthName
DateTimeFormatInfo.GetAllDateTimePatterns
DateTimeFormatInfo.GetDayName
DateTimeFormatInfo.GetEra
DateTimeFormatInfo.GetEraName
DateTimeFormatInfo.GetFormat
DateTimeFormatInfo.GetInstance
DateTimeFormatInfo.GetMonthName
DateTimeFormatInfo.ReadOnly
[DateTimeFormatInfo Properties]
DateTimeFormatInfo.AbbreviatedDayNames
DateTimeFormatInfo.AbbreviatedMonthGenitiveNames
DateTimeFormatInfo.AbbreviatedMonthNames
DateTimeFormatInfo.AMDesignator
DateTimeFormatInfo.Calendar
DateTimeFormatInfo.CalendarWeekRule
DateTimeFormatInfo.CurrentInfo
DateTimeFormatInfo.DateSeparator
DateTimeFormatInfo.DayNames
DateTimeFormatInfo.FirstDayOfWeek
DateTimeFormatInfo.FullDateTimePattern
DateTimeFormatInfo.InvariantInfo
DateTimeFormatInfo.IsReadOnly
DateTimeFormatInfo.LongDatePattern
DateTimeFormatInfo.LongTimePattern
DateTimeFormatInfo.MonthDayPattern
DateTimeFormatInfo.MonthGenitiveNames
DateTimeFormatInfo.MonthNames
DateTimeFormatInfo.PMDesignator
DateTimeFormatInfo.RFC1123Pattern
DateTimeFormatInfo.ShortDatePattern
DateTimeFormatInfo.ShortestDayNames
DateTimeFormatInfo.ShortTimePattern
DateTimeFormatInfo.SortableDateTimePattern
DateTimeFormatInfo.TimeSeparator
DateTimeFormatInfo.UniversalSortableDateTimePattern
DateTimeFormatInfo.YearMonthPattern

System.Globalization.DateTimeStyles enumeration

System.Globalization.DaylightTime class

DaylightTime.DaylightTime
[DaylightTime Properties]

DaylightTime.Delta
DaylightTime.End
DaylightTime.Start

System.Globalization.GregorianCalendar class

GregorianCalendar.GregorianCalendar
[GregorianCalendar Fields]
GregorianCalendar.ADEra
[GregorianCalendar Methods]
GregorianCalendar.AddMonths
GregorianCalendar.AddYears
GregorianCalendar.GetDayOfMonth
GregorianCalendar.GetDayOfWeek
GregorianCalendar.GetDayOfYear
GregorianCalendar.GetDaysInMonth
GregorianCalendar.GetDaysInYear
GregorianCalendar.GetEra
GregorianCalendar.GetMonth
GregorianCalendar.GetMonthsInYear
GregorianCalendar.GetWeekOfYear
GregorianCalendar.GetYear
GregorianCalendar.IsLeapDay
GregorianCalendar.IsLeapMonth
GregorianCalendar.IsLeapYear
GregorianCalendar.ToDateTime
GregorianCalendar.ToFourDigitYear
[GregorianCalendar Properties]
GregorianCalendar.CalendarType
GregorianCalendar.Eras
GregorianCalendar.MaxSupportedDateTime
GregorianCalendar.MinSupportedDateTime
GregorianCalendar.TwoDigitYearMax

System.Globalization.GregorianCalendarTypes enumeration

System.Globalization.JapaneseCalendar class

JapaneseCalendar.JapaneseCalendar
[JapaneseCalendar Methods]
JapaneseCalendar.AddMonths
JapaneseCalendar.AddYears
JapaneseCalendar.GetDayOfMonth
JapaneseCalendar.GetDayOfWeek
JapaneseCalendar.GetDayOfYear
JapaneseCalendar.GetDaysInMonth
JapaneseCalendar.GetDaysInYear
JapaneseCalendar.GetEra
JapaneseCalendar.GetMonth
JapaneseCalendar.GetMonthsInYear
JapaneseCalendar.GetWeekOfYear
JapaneseCalendar.GetYear
JapaneseCalendar.IsLeapDay
JapaneseCalendar.IsLeapMonth
JapaneseCalendar.IsLeapYear
JapaneseCalendar.ToDateTime
JapaneseCalendar.ToFourDigitYear
[JapaneseCalendar Properties]
JapaneseCalendar.Eras
JapaneseCalendar.MaxSupportedDateTime
JapaneseCalendar.MinSupportedDateTime
JapaneseCalendar.TwoDigitYearMax

System.Globalization.KoreanCalendar class

KoreanCalendar.KoreanCalendar
[KoreanCalendar Fields]
KoreanCalendar.KoreanEra
[KoreanCalendar Methods]
KoreanCalendar.AddMonths
KoreanCalendar.AddYears
KoreanCalendar.GetDayOfMonth
KoreanCalendar.GetDayOfWeek
KoreanCalendar.GetDayOfYear
KoreanCalendar.GetDaysInMonth
KoreanCalendar.GetDaysInYear
KoreanCalendar.GetEra
KoreanCalendar.GetMonth
KoreanCalendar.GetMonthsInYear
KoreanCalendar.GetWeekOfYear
KoreanCalendar.GetYear
KoreanCalendar.IsLeapDay
KoreanCalendar.IsLeapMonth
KoreanCalendar.IsLeapYear
KoreanCalendar.ToDateTime
KoreanCalendar.ToFourDigitYear
[KoreanCalendar Properties]
KoreanCalendar.Eras
KoreanCalendar.MaxSupportedDateTime
KoreanCalendar.MinSupportedDateTime
KoreanCalendar.TwoDigitYearMax

System.Globalization.NumberFormatInfo class

NumberFormatInfo.NumberFormatInfo
[NumberFormatInfo Methods]
NumberFormatInfo.Clone
NumberFormatInfo.GetFormat
NumberFormatInfo.GetInstance
NumberFormatInfo.ReadOnly
[NumberFormatInfo Properties]
NumberFormatInfo.CurrencyDecimalDigits
NumberFormatInfo.CurrencyDecimalSeparator
NumberFormatInfo.CurrencyGroupSeparator
NumberFormatInfo.CurrencyGroupSizes
NumberFormatInfo.CurrencyNegativePattern
NumberFormatInfo.CurrencyPositivePattern
NumberFormatInfo.CurrencySymbol
NumberFormatInfo.CurrentInfo
NumberFormatInfo.InvariantInfo
NumberFormatInfo.IsReadOnly
NumberFormatInfo.NaNSymbol
NumberFormatInfo.NegativeInfinitySymbol
NumberFormatInfo.NegativeSign
NumberFormatInfo.NumberDecimalDigits
NumberFormatInfo.NumberDecimalSeparator
NumberFormatInfo.NumberGroupSeparator
NumberFormatInfo.NumberGroupSizes
NumberFormatInfo.NumberNegativePattern
NumberFormatInfo.PercentDecimalDigits
NumberFormatInfo.PercentDecimalSeparator
NumberFormatInfo.PercentGroupSeparator
NumberFormatInfo.PercentGroupSizes
NumberFormatInfo.PercentNegativePattern
NumberFormatInfo.PercentPositivePattern
NumberFormatInfo.PercentSymbol
NumberFormatInfo.PerMilleSymbol
NumberFormatInfo.PositiveInfinitySymbol

NumberFormatInfo.PositiveSign

System.Globalization.NumberStyles enumeration

System.Globalization.RegionInfo class

RegionInfo.RegionInfo

[RegionInfo Methods]

RegionInfo.Equals

RegionInfo.GetHashCode

RegionInfo.ToString

[RegionInfo Properties]

RegionInfo.CurrencySymbol

RegionInfo.CurrentRegion

RegionInfo.EnglishName

RegionInfo.IsMetric

RegionInfo.ISOCurrencySymbol

RegionInfo.Name

RegionInfo.ThreeLetterISORegionName

RegionInfo.ThreeLetterWindowsRegionName

RegionInfo.TwoLetterISORegionName

System.Globalization.StringInfo class

StringInfo.StringInfo

[StringInfo Methods]

StringInfo.Equals

StringInfo.GetHashCode

StringInfo.GetNextTextElement

StringInfo.GetTextElementEnumerator

StringInfo.ParseCombiningCharacters

[StringInfo Properties]

StringInfo.LengthInTextElements

StringInfo.String

System.Globalization.TaiwanCalendar class

TaiwanCalendar.TaiwanCalendar

[TaiwanCalendar Methods]

TaiwanCalendar.AddMonths

TaiwanCalendar.AddYears

TaiwanCalendar.GetDayOfMonth

TaiwanCalendar.GetDayOfWeek

TaiwanCalendar.GetDayOfYear

TaiwanCalendar.GetDaysInMonth

TaiwanCalendar.GetDaysInYear

TaiwanCalendar.GetEra

TaiwanCalendar.GetMonth

TaiwanCalendar.GetMonthsInYear

TaiwanCalendar.GetWeekOfYear

TaiwanCalendar.GetYear

TaiwanCalendar.IsLeapDay

TaiwanCalendar.IsLeapMonth

TaiwanCalendar.IsLeapYear

TaiwanCalendar.ToDateTime

TaiwanCalendar.ToFourDigitYear

[TaiwanCalendar Properties]

TaiwanCalendar.Eras

TaiwanCalendar.MaxSupportedDateTime

TaiwanCalendar.MinSupportedDateTime

TaiwanCalendar.TwoDigitYearMax

System.Globalization.TextElementEnumerator class

[TextElementEnumerator Methods]

TextElementEnumerator.GetTextElement

- TextElementEnumerator.MoveNext
- TextElementEnumerator.Reset
- [TextElementEnumerator Properties]
- TextElementEnumerator.Current
- TextElementEnumerator.ElementIndex

System.Globalization.TextInfo class

- [TextInfo Methods]
- TextInfo.Equals
- TextInfo.GetHashCode
- TextInfo.ToLower
- TextInfo.ToString
- TextInfo.ToTitleCase
- TextInfo.ToUpper
- [TextInfo Properties]
- TextInfo.ANSICodePage
- TextInfo.CultureName
- TextInfo.IsReadOnly
- TextInfo.ListSeparator

System.Globalization.ThaiBuddhistCalendar class

- ThaiBuddhistCalendar.ThaiBuddhistCalendar
- [ThaiBuddhistCalendar Fields]
- ThaiBuddhistCalendar.ThaiBuddhistEra
- [ThaiBuddhistCalendar Methods]
- ThaiBuddhistCalendar.AddMonths
- ThaiBuddhistCalendar.AddYears
- ThaiBuddhistCalendar.GetDayOfMonth
- ThaiBuddhistCalendar.GetDayOfWeek
- ThaiBuddhistCalendar.GetDayOfYear
- ThaiBuddhistCalendar.GetDaysInMonth
- ThaiBuddhistCalendar.GetDaysInYear
- ThaiBuddhistCalendar.GetEra
- ThaiBuddhistCalendar.GetMonth
- ThaiBuddhistCalendar.GetMonthsInYear
- ThaiBuddhistCalendar.GetWeekOfYear
- ThaiBuddhistCalendar.GetYear
- ThaiBuddhistCalendar.IsLeapDay
- ThaiBuddhistCalendar.IsLeapMonth
- ThaiBuddhistCalendar.IsLeapYear
- ThaiBuddhistCalendar.ToDateTime
- ThaiBuddhistCalendar.ToFourDigitYear
- [ThaiBuddhistCalendar Properties]
- ThaiBuddhistCalendar.Eras
- ThaiBuddhistCalendar.MaxSupportedDateTime
- ThaiBuddhistCalendar.MinSupportedDateTime
- ThaiBuddhistCalendar.TwoDigitYearMax

System.Globalization.UnicodeCategory enumeration

System.IO namespace

System.IO.BinaryReader class

- BinaryReader.BinaryReader
- [BinaryReader Methods]
- BinaryReader.Close
- BinaryReader.Dispose
- BinaryReader.FillBuffer
- BinaryReader.PeekChar
- BinaryReader.Read
- BinaryReader.Read7BitEncodedInt

BinaryReader.ReadBoolean
BinaryReader.ReadByte
BinaryReader.ReadBytes
BinaryReader.ReadChar
BinaryReader.ReadChars
BinaryReader.ReadDouble
BinaryReader.ReadInt16
BinaryReader.ReadInt32
BinaryReader.ReadInt64
BinaryReader.ReadSByte
BinaryReader.ReadSingle
BinaryReader.ReadString
BinaryReader.ReadUInt16
BinaryReader.ReadUInt32
BinaryReader.ReadUInt64
[BinaryReader Properties]
BinaryReader.BaseStream

System.IO.BinaryWriter class

BinaryWriter.BinaryWriter
[BinaryWriter Fields]
BinaryWriter.Null
BinaryWriter.OutputStream
[BinaryWriter Methods]
BinaryWriter.Close
BinaryWriter.Dispose
BinaryWriter.Flush
BinaryWriter.Seek
BinaryWriter.Write
BinaryWriter.Write7BitEncodedInt
[BinaryWriter Properties]
BinaryWriter.BaseStream

System.IO.Directory class

[Directory Methods]
Directory.CreateDirectory
Directory.Delete
Directory.Exists
Directory.GetCreationTime
Directory.GetCurrentDirectory
Directory.GetDirectories
Directory.GetDirectoryRoot
Directory.GetFiles
Directory.GetFileSystemEntries
Directory.GetLastAccessTime
Directory.GetLastWriteTime
Directory.Move
Directory.SetCurrentDirectory

System.IO.DirectoryInfo class

DirectoryInfo.DirectoryInfo
[DirectoryInfo Methods]
DirectoryInfo.Create
DirectoryInfo.CreateSubdirectory
DirectoryInfo.Delete
DirectoryInfo.GetDirectories
DirectoryInfo.GetFiles
DirectoryInfo.GetFileSystemInfos
DirectoryInfo.MoveTo
DirectoryInfo.ToString
[DirectoryInfo Properties]
DirectoryInfo.Exists

DirectoryInfo.Name
DirectoryInfo.Parent
DirectoryInfo.Root

System.IO.DirectoryNotFoundException class
DirectoryNotFoundException.DirectoryNotFoundException

System.IO.EndOfStreamException class
EndOfStreamException.EndOfStreamException

System.IO.File class
[File Methods]

File.AppendText
File.Copy
File.Create
File.CreateText
File.Delete
File.Exists
File.GetCreationTime
File.GetLastAccessTime
File.GetLastWriteTime
File.Move
File.Open
File.OpenRead
File.OpenText
File.OpenWrite

System.IO.FileAccess enumeration

System.IO.FileAttributes enumeration

System.IO.FileInfo class

FileInfo.FileInfo
[FileInfo Methods]
FileInfo.AppendText
FileInfo.CopyTo
FileInfo.Create
FileInfo.CreateText
FileInfo.Delete
FileInfo.MoveTo
FileInfo.Open
FileInfo.OpenRead
FileInfo.OpenText
FileInfo.OpenWrite
FileInfo.ToString
[FileInfo Properties]
FileInfo.Directory
FileInfo.DirectoryName
FileInfo.Exists
FileInfo.Length
FileInfo.Name

System.IO.FileMode enumeration

System.IO.FileNotFoundException class
FileNotFoundException.FileNotFoundException

System.IO.FileShare enumeration

System.IO.FileStream class
FileStream.FileStream
[FileStream Methods]

FileStream.Dispose
FileStream.Finalize
FileStream.Flush
FileStream.Read
FileStream.ReadByte
FileStream.Seek
FileStream.SetLength
FileStream.Write
FileStream.WriteByte
[FileStream Properties]
FileStream.CanRead
FileStream.CanSeek
FileStream.CanWrite
FileStream.IsAsync
FileStream.Length
FileStream.Name
FileStream.Position

System.IO.FileSystemInfo class

FileSystemInfo.FileSystemInfo
[FileSystemInfo Fields]
FileSystemInfo.FullPath
FileSystemInfo.OriginalPath
[FileSystemInfo Methods]
FileSystemInfo.Delete
FileSystemInfo.Refresh
[FileSystemInfo Properties]
FileSystemInfo.Attributes
FileSystemInfo.CreationTime
FileSystemInfo.Exists
FileSystemInfo.Extension
FileSystemInfo.FullName
FileSystemInfo.LastAccessTime
FileSystemInfo.LastWriteTime
FileSystemInfo.Name

System.IO.IOException class

IOException.IOException

System.IO.MemoryStream class

MemoryStream.MemoryStream
[MemoryStream Methods]
MemoryStream.Dispose
MemoryStream.Flush
MemoryStream.GetBuffer
MemoryStream.Read
MemoryStream.ReadByte
MemoryStream.Seek
MemoryStream.SetLength
MemoryStream.ToArray
MemoryStream.Write
MemoryStream.WriteByte
MemoryStream.WriteTo
[MemoryStream Properties]
MemoryStream.CanRead
MemoryStream.CanSeek
MemoryStream.CanWrite
MemoryStream.Capacity
MemoryStream.Length
MemoryStream.Position

System.IO.Path class

[Path Fields]

- Path.AltDirectorySeparatorChar
- Path.DirectorySeparatorChar
- Path.InvalidPathChars
- Path.PathSeparator
- Path.VolumeSeparatorChar

[Path Methods]

- Path.ChangeExtension
- Path.Combine
- Path.GetDirectoryName
- Path.GetExtension
- Path.GetFileName
- Path.GetFileNameWithoutExtension
- Path.GetFullPath
- Path.GetInvalidPathChars
- Path.GetPathRoot
- Path.HasExtension
- Path.IsPathRooted

System.IO.PathTooLongException class

- PathTooLongException.PathTooLongException

System.IO.SeekOrigin enumeration

System.IO.Stream class

- Stream.Stream

[Stream Fields]

- Stream.Null

[Stream Methods]

- Stream.BeginRead
- Stream.BeginWrite
- Stream.Close
- Stream.CreateWaitHandle
- Stream.Dispose
- Stream.EndRead
- Stream.EndWrite
- Stream.Flush
- Stream.Read
- Stream.ReadByte
- Stream.Seek
- Stream.SetLength
- Stream.Write
- Stream.WriteByte

[Stream Properties]

- Stream.CanRead
- Stream.CanSeek
- Stream.CanTimeout
- Stream.CanWrite
- Stream.Length
- Stream.Position
- Stream.ReadTimeout
- Stream.WriteTimeout

System.IO.StreamReader class

- StreamReader.StreamReader

[StreamReader Fields]

- StreamReader.Null

[StreamReader Methods]

- StreamReader.Close
- StreamReader.DiscardBufferedData
- StreamReader.Dispose
- StreamReader.Peek

- StreamReader.Read
- StreamReader.ReadLine
- StreamReader.ReadToEnd
- [StreamReader Properties]
- StreamReader.BaseStream
- StreamReader.CurrentEncoding
- StreamReader.EndOfStream

System.IO.StreamWriter class

- StreamWriter.StreamWriter
- [StreamWriter Fields]
- StreamWriter.Null
- [StreamWriter Methods]
- StreamWriter.Close
- StreamWriter.Dispose
- StreamWriter.Flush
- StreamWriter.Write
- [StreamWriter Properties]
- StreamWriter.AutoFlush
- StreamWriter.BaseStream
- StreamWriter.Encoding

System.IO.StringReader class

- StringReader.StringReader
- [StringReader Methods]
- StringReader.Close
- StringReader.Dispose
- StringReader.Peek
- StringReader.Read
- StringReader.ReadLine
- StringReader.ReadToEnd

System.IO.StringWriter class

- StringWriter.StringWriter
- [StringWriter Methods]
- StringWriter.Close
- StringWriter.Dispose
- StringWriter.GetStringBuilder
- StringWriter.ToString
- StringWriter.Write
- [StringWriter Properties]
- StringWriter.Encoding

System.IO.TextReader class

- TextReader.TextReader
- [TextReader Fields]
- TextReader.Null
- [TextReader Methods]
- TextReader.Close
- TextReader.Dispose
- TextReader.Peek
- TextReader.Read
- TextReader.ReadBlock
- TextReader.ReadLine
- TextReader.ReadToEnd
- TextReader.Synchronized

System.IO.TextWriter class

- TextWriter.TextWriter
- [TextWriter Fields]
- TextWriter.NewLine
- TextWriter.Null

- [TextWriter Methods]
 - TextWriter.Close
 - TextWriter.Dispose
 - TextWriter.Flush
 - TextWriter.Synchronized
 - TextWriter.Write
 - TextWriter.WriteLine
- [TextWriter Properties]
 - TextWriter.Encoding
 - TextWriter.FormatProvider
 - TextWriter.NewLine

System.Net namespace

System.Net.ICredentials interface

- [ICredentials Methods]
 - ICredentials.GetCredential

System.Net.NetworkCredential class

- NetworkCredential.NetworkCredential
- [NetworkCredential Methods]
 - NetworkCredential.GetCredential
- [NetworkCredential Properties]
 - NetworkCredential.Domain
 - NetworkCredential.Password
 - NetworkCredential.UserName

System.Reflection namespace

System.Reflection.AmbiguousMatchException class

- AmbiguousMatchException.AmbiguousMatchException

System.Reflection.Assembly class

- [Assembly Methods]
 - Assembly.CreateInstance
 - Assembly.GetCallingAssembly
 - Assembly.GetCustomAttributes
 - Assembly.GetExecutingAssembly
 - Assembly.GetManifestResourceNames
 - Assembly.GetManifestResourceStream
 - Assembly.GetModules
 - Assembly.GetName
 - Assembly.GetSatelliteAssembly
 - Assembly.GetType
 - Assembly.GetTypes
 - Assembly.IsDefined
 - Assembly.Load
 - Assembly.LoadFrom
 - Assembly.ToString
- [Assembly Properties]
 - Assembly.FullName
 - Assembly.ManifestModule

System.Reflection.AssemblyAlgorithmIdAttribute class

- AssemblyAlgorithmIdAttribute.AssemblyAlgorithmIdAttribute
- [AssemblyAlgorithmIdAttribute Properties]
 - AssemblyAlgorithmIdAttribute.AlgorithmId

System.Reflection.AssemblyCompanyAttribute class

- AssemblyCompanyAttribute.AssemblyCompanyAttribute

[AssemblyCompanyAttribute Properties]
AssemblyCompanyAttribute.Company

System.Reflection.AssemblyConfigurationAttribute class
AssemblyConfigurationAttribute.AssemblyConfigurationAttribute
[AssemblyConfigurationAttribute Properties]
AssemblyConfigurationAttribute.Configuration

System.Reflection.AssemblyCopyrightAttribute class
AssemblyCopyrightAttribute.AssemblyCopyrightAttribute
[AssemblyCopyrightAttribute Properties]
AssemblyCopyrightAttribute.Copyright

System.Reflection.AssemblyCultureAttribute class
AssemblyCultureAttribute.AssemblyCultureAttribute
[AssemblyCultureAttribute Properties]
AssemblyCultureAttribute.Culture

System.Reflection.AssemblyDefaultAliasAttribute class
AssemblyDefaultAliasAttribute.AssemblyDefaultAliasAttribute
[AssemblyDefaultAliasAttribute Properties]
AssemblyDefaultAliasAttribute.DefaultAlias

System.Reflection.AssemblyDelaySignAttribute class
AssemblyDelaySignAttribute.AssemblyDelaySignAttribute
[AssemblyDelaySignAttribute Properties]
AssemblyDelaySignAttribute.DelaySign

System.Reflection.AssemblyDescriptionAttribute class
AssemblyDescriptionAttribute.AssemblyDescriptionAttribute
[AssemblyDescriptionAttribute Properties]
AssemblyDescriptionAttribute.Description

System.Reflection.AssemblyFlagsAttribute class
AssemblyFlagsAttribute.AssemblyFlagsAttribute
[AssemblyFlagsAttribute Properties]
AssemblyFlagsAttribute.AssemblyFlags
AssemblyFlagsAttribute.Flags

System.Reflection.AssemblyInformationalVersionAttribute class
AssemblyInformationalVersionAttribute.AssemblyInformationalVersionAttribute
[AssemblyInformationalVersionAttribute Properties]
AssemblyInformationalVersionAttribute.InformationalVersion

System.Reflection.AssemblyKeyFileAttribute class
AssemblyKeyFileAttribute.AssemblyKeyFileAttribute
[AssemblyKeyFileAttribute Properties]
AssemblyKeyFileAttribute.KeyFile

System.Reflection.AssemblyKeyNameAttribute class
AssemblyKeyNameAttribute.AssemblyKeyNameAttribute
[AssemblyKeyNameAttribute Properties]
AssemblyKeyNameAttribute.KeyName

System.Reflection.AssemblyName class
AssemblyName.AssemblyName
[AssemblyName Methods]
AssemblyName.Clone
AssemblyName.GetPublicKey
AssemblyName.GetPublicKeyToken
AssemblyName.SetPublicKey
AssemblyName.SetPublicKeyToken

- AssemblyName.ToString
- [AssemblyName Properties]
- AssemblyName.CodeBase
- AssemblyName.CultureInfo
- AssemblyName.Flags
- AssemblyName.FullName
- AssemblyName.HashAlgorithm
- AssemblyName.Name
- AssemblyName.Version
- AssemblyName.VersionCompatibility

System.Reflection.AssemblyNameFlags enumeration

System.Reflection.AssemblyProductAttribute class

- AssemblyProductAttribute.AssemblyProductAttribute
- [AssemblyProductAttribute Properties]
- AssemblyProductAttribute.Product

System.Reflection.AssemblyTitleAttribute class

- AssemblyTitleAttribute.AssemblyTitleAttribute
- [AssemblyTitleAttribute Properties]
- AssemblyTitleAttribute.Title

System.Reflection.AssemblyTrademarkAttribute class

- AssemblyTrademarkAttribute.AssemblyTrademarkAttribute
- [AssemblyTrademarkAttribute Properties]
- AssemblyTrademarkAttribute.Trademark

System.Reflection.AssemblyVersionAttribute class

- AssemblyVersionAttribute.AssemblyVersionAttribute
- [AssemblyVersionAttribute Properties]
- AssemblyVersionAttribute.Version

System.Reflection.Binder class

- Binder.Binder
- [Binder Methods]
- Binder.BindToField
- Binder.BindToMethod
- Binder.ChangeType
- Binder.ReorderArgumentArray
- Binder.SelectMethod
- Binder.SelectProperty

System.Reflection.BindingFlags enumeration

System.Reflection.CallingConventions enumeration

System.Reflection.ConstructorInfo class

- ConstructorInfo.ConstructorInfo
- [ConstructorInfo Fields]
- ConstructorInfo.ConstructorName
- ConstructorInfo.TypeConstructorName
- [ConstructorInfo Methods]
- ConstructorInfo.Invoke
- [ConstructorInfo Properties]
- ConstructorInfo.MemberType

System.Reflection.CustomAttributeFormatException class

- CustomAttributeFormatException.CustomAttributeFormatException

System.Reflection.DefaultMemberAttribute class

- DefaultMemberAttribute.DefaultMemberAttribute

[DefaultMemberAttribute Properties]
DefaultMemberAttribute.MemberName

System.Reflection.EventAttributes enumeration

System.Reflection.EventInfo class

EventInfo.EventInfo
[EventInfo Methods]
EventInfo.AddEventHandler
EventInfo.GetAddMethod
EventInfo.GetRaiseMethod
EventInfo.GetRemoveMethod
EventInfo.RemoveEventHandler
[EventInfo Properties]
EventInfo.Attributes
EventInfo.EventHandlerType
EventInfo.IsMulticast
EventInfo.IsSpecialName
EventInfo.MemberType

System.Reflection.FieldAttributes enumeration

System.Reflection.FieldInfo class

FieldInfo.FieldInfo
[FieldInfo Methods]
FieldInfo.GetFieldFromHandle
FieldInfo.GetValue
FieldInfo.SetValue
[FieldInfo Properties]
FieldInfo.Attributes
FieldInfo.FieldHandle
FieldInfo.FieldType
FieldInfo.IsAssembly
FieldInfo.IsFamily
FieldInfo.IsFamilyAndAssembly
FieldInfo.IsFamilyOrAssembly
FieldInfo.IsInitOnly
FieldInfo.IsLiteral
FieldInfo.IsNotSerialized
FieldInfo.IsPInvokeImpl
FieldInfo.IsPrivate
FieldInfo.IsPublic
FieldInfo.IsSpecialName
FieldInfo.IsStatic
FieldInfo.MemberType

System.Reflection.ICustomAttributeProvider interface

[ICustomAttributeProvider Methods]
ICustomAttributeProvider.GetCustomAttributes
ICustomAttributeProvider.IsDefined

System.Reflection.MemberInfo class

MemberInfo.MemberInfo
[MemberInfo Methods]
MemberInfo.GetCustomAttributes
MemberInfo.IsDefined
[MemberInfo Properties]
MemberInfo.DeclaringType
MemberInfo.MemberType
MemberInfo.Name
MemberInfo.ReflectedType

System.Reflection.MemberTypes enumeration

System.Reflection.MethodAttributes enumeration

System.Reflection.MethodBase class

MethodBase.MethodBase

[MethodBase Methods]

MethodBase.GetGenericArguments

MethodBase.GetMethodFromHandle

MethodBase.GetParameters

MethodBase.Invoke

[MethodBase Properties]

MethodBase.Attributes

MethodBase.CallingConvention

MethodBase.ContainsGenericParameters

MethodBase.IsAbstract

MethodBase.IsAssembly

MethodBase.IsConstructor

MethodBase.IsFamily

MethodBase.IsFamilyAndAssembly

MethodBase.IsFamilyOrAssembly

MethodBase.IsFinal

MethodBase.IsGenericMethod

MethodBase.IsGenericMethodDefinition

MethodBase.IsHideBySig

MethodBase.IsPrivate

MethodBase.IsPublic

MethodBase.IsSpecialName

MethodBase.IsStatic

MethodBase.IsVirtual

MethodBase.MethodHandle

System.Reflection.MethodImplAttributes enumeration

System.Reflection.MethodInfo class

MethodInfo.MethodInfo

[MethodInfo Methods]

MethodInfo.GetBaseDefinition

MethodInfo.GetGenericArguments

MethodInfo.GetGenericMethodDefinition

MethodInfo.MakeGenericMethod

[MethodInfo Properties]

MethodInfo.ContainsGenericParameters

MethodInfo.IsGenericMethod

MethodInfo.IsGenericMethodDefinition

MethodInfo.MemberType

MethodInfo.ReturnType

MethodInfo.ReturnTypeCustomAttributes

System.Reflection.Missing class

[Missing Fields]

Missing.Value

System.Reflection.Module class

[Module Methods]

Module.GetCustomAttributes

Module.GetType

Module.GetTypes

Module.IsDefined

Module.ToString

[Module Properties]

Module.Assembly

Module.FullyQualifiedName
Module.Name

System.Reflection.ParameterAttributes enumeration

System.Reflection.ParameterInfo class

ParameterInfo.ParameterInfo
[ParameterInfo Fields]
ParameterInfo.AttrsImpl
ParameterInfo.ClassImpl
ParameterInfo.DefaultValueImpl
ParameterInfo.MemberImpl
ParameterInfo.NameImpl
ParameterInfo.PositionImpl
[ParameterInfo Methods]
ParameterInfo.Equals
ParameterInfo.GetCustomAttributes
ParameterInfo.GetHashCode
ParameterInfo.IsDefined
[ParameterInfo Properties]
ParameterInfo.Attributes
ParameterInfo.DefaultValue
ParameterInfo.Member
ParameterInfo.Name
ParameterInfo.ParameterType
ParameterInfo.Position

System.Reflection.ParameterModifier structure

ParameterModifier.ParameterModifier
[ParameterModifier Properties]
ParameterModifier.Item

System.Reflection.PropertyAttributes enumeration

System.Reflection.PropertyInfo class

PropertyInfo.PropertyInfo
[PropertyInfo Methods]
PropertyInfo.GetAccessors
PropertyInfo.GetGetMethod
PropertyInfo.GetIndexParameters
PropertyInfo.GetSetMethod
PropertyInfo.GetValue
PropertyInfo.SetValue
[PropertyInfo Properties]
PropertyInfo.Attributes
PropertyInfo.CanRead
PropertyInfo.CanWrite
PropertyInfo.IsSpecialName
PropertyInfo.MemberType
PropertyInfo.PropertyType

System.Reflection.TargetInvocationException class

TargetInvocationException.TargetInvocationException

System.Reflection.TargetParameterCountException class

TargetParameterCountException.TargetParameterCountException

System.Reflection.TypeAttributes enumeration

System.Resources namespace

System.Resources.IResourceReader interface

[IResourceReader Methods]

IResourceReader.Close

IResourceReader.GetEnumerator

System.Resources.MissingManifestResourceException class

MissingManifestResourceException.MissingManifestResourceException

System.Resources.NeutralResourcesLanguageAttribute class

NeutralResourcesLanguageAttribute.NeutralResourcesLanguageAttribute

[NeutralResourcesLanguageAttribute Properties]

NeutralResourcesLanguageAttribute.CultureName

System.Resources.ResourceManager class

ResourceManager.ResourceManager

[ResourceManager Fields]

ResourceManager.BaseNameField

ResourceManager.HeaderVersionNumber

ResourceManager.MagicNumber

ResourceManager.MainAssembly

ResourceManager.ResourceSets

[ResourceManager Methods]

ResourceManager.CreateFileBasedResourceManager

ResourceManager.GetNeutralResourcesLanguage

ResourceManager.GetObject

ResourceManager.GetResourceFileName

ResourceManager.GetResourceSet

ResourceManager.GetSatelliteContractVersion

ResourceManager.GetString

ResourceManager.InternalGetResourceSet

ResourceManager.ReleaseAllResources

[ResourceManager Properties]

ResourceManager.BaseName

ResourceManager.IgnoreCase

ResourceManager.ResourceSetType

System.Resources.ResourceReader class

ResourceReader.ResourceReader

[ResourceReader Methods]

ResourceReader.Close

ResourceReader.GetEnumerator

System.Resources.ResourceSet class

ResourceSet.ResourceSet

[ResourceSet Fields]

ResourceSet.Reader

ResourceSet.Table

[ResourceSet Methods]

ResourceSet.Close

ResourceSet.Dispose

ResourceSet.GetDefaultReader

ResourceSet.GetEnumerator

ResourceSet.GetObject

ResourceSet.GetString

ResourceSet.ReadResources

System.Resources.SatelliteContractVersionAttribute class

SatelliteContractVersionAttribute.SatelliteContractVersionAttribute

[SatelliteContractVersionAttribute Properties]

SatelliteContractVersionAttribute.Version

System.Runtime.CompilerServices namespace

System.Runtime.CompilerServices.AccessedThroughPropertyAttribute class

AccessedThroughPropertyAttribute.AccessedThroughPropertyAttribute
[AccessedThroughPropertyAttribute Properties]
AccessedThroughPropertyAttribute.PropertyName

System.Runtime.CompilerServices.CompilerGeneratedAttribute class

CompilerGeneratedAttribute.CompilerGeneratedAttribute

System.Runtime.CompilerServices.CustomConstantAttribute class

CustomConstantAttribute.CustomConstantAttribute
[CustomConstantAttribute Properties]
CustomConstantAttribute.Value

System.Runtime.CompilerServices.DateTimeConstantAttribute class

DateTimeConstantAttribute.DateTimeConstantAttribute
[DateTimeConstantAttribute Properties]
DateTimeConstantAttribute.Value

System.Runtime.CompilerServices.DecimalConstantAttribute class

DecimalConstantAttribute.DecimalConstantAttribute
[DecimalConstantAttribute Properties]
DecimalConstantAttribute.Value

System.Runtime.CompilerServices.FixedBufferAttribute class

FixedBufferAttribute.FixedBufferAttribute
[FixedBufferAttribute Properties]
FixedBufferAttribute.ElementType
FixedBufferAttribute.Length

System.Runtime.CompilerServices.IndexerNameAttribute class

IndexerNameAttribute.IndexerNameAttribute

System.Runtime.CompilerServices.InternalsVisibleToAttribute class

InternalsVisibleToAttribute.InternalsVisibleToAttribute
[InternalsVisibleToAttribute Properties]
InternalsVisibleToAttribute.AssemblyName

System.Runtime.CompilerServices.IsVolatile class

System.Runtime.CompilerServices.MethodCodeType enumeration

System.Runtime.CompilerServices.MethodImplAttribute class

MethodImplAttribute.MethodImplAttribute
[MethodImplAttribute Fields]
MethodImplAttribute.MethodCodeType
[MethodImplAttribute Properties]
MethodImplAttribute.Value

System.Runtime.CompilerServices.MethodImplOptions enumeration

System.Runtime.CompilerServices.RuntimeHelpers class

[RuntimeHelpers Methods]
RuntimeHelpers.GetObjectValue
RuntimeHelpers.InitializeArray
[RuntimeHelpers Properties]
RuntimeHelpers.OffsetToStringData

System.Runtime.InteropServices namespace

System.Runtime.InteropServices.CallingConvention enumeration

System.Runtime.InteropServices.CharSet enumeration

System.Runtime.InteropServices.ComVisibleAttribute class

ComVisibleAttribute.ComVisibleAttribute
[ComVisibleAttribute Properties]
ComVisibleAttribute.Value

System.Runtime.InteropServices.DllImportAttribute class

DllImportAttribute.DllImportAttribute
[DllImportAttribute Fields]
DllImportAttribute.CallingConvention
DllImportAttribute.CharSet
DllImportAttribute.EntryPoint
DllImportAttribute.PreserveSig
DllImportAttribute.SetLastError
[DllImportAttribute Properties]
DllImportAttribute.Value

System.Runtime.InteropServices.ExternalException class

ExternalException.ExternalException
[ExternalException Properties]
ExternalException.ErrorCode

System.Runtime.InteropServices.FieldOffsetAttribute class

FieldOffsetAttribute.FieldOffsetAttribute
[FieldOffsetAttribute Properties]
FieldOffsetAttribute.Value

System.Runtime.InteropServices.GCHandle structure

[GCHandle Methods]
GCHandle.AddrOfPinnedObject
GCHandle.Alloc
GCHandle.Free
GCHandle.op_Explicit
[GCHandle Properties]
GCHandle.IsAllocated
GCHandle.Target

System.Runtime.InteropServices.GCHandleType enumeration

System.Runtime.InteropServices.GuidAttribute class

GuidAttribute.GuidAttribute
[GuidAttribute Properties]
GuidAttribute.Value

System.Runtime.InteropServices.InAttribute class

InAttribute.InAttribute

System.Runtime.InteropServices.LayoutKind enumeration

System.Runtime.InteropServices.Marshal class

[Marshal Fields]
Marshal.SystemDefaultCharSize
[Marshal Methods]
Marshal.Copy
Marshal.GetLastWin32Error
Marshal.Prelink
Marshal.PrelinkAll
Marshal.PtrToStringUni
Marshal.PtrToStructure

Marshal.ReadByte
Marshal.ReadInt16
Marshal.ReadInt32
Marshal.ReadInt64
Marshal.ReadIntPtr
Marshal.SizeOf
Marshal.StructureToPtr
Marshal.WriteByte
Marshal.WriteInt16
Marshal.WriteInt32
Marshal.WriteInt64
Marshal.WriteIntPtr

System.Runtime.InteropServices.MarshalAsAttribute class

MarshalAsAttribute.MarshalAsAttribute
[MarshalAsAttribute Fields]
MarshalAsAttribute.ArraySubType
MarshalAsAttribute.SizeConst
MarshalAsAttribute.SizeParamIndex
[MarshalAsAttribute Properties]
MarshalAsAttribute.Value

System.Runtime.InteropServices.MarshalDirectiveException class

MarshalDirectiveException.MarshalDirectiveException

System.Runtime.InteropServices.OutAttribute class

OutAttribute.OutAttribute

System.Runtime.InteropServices.PreserveSigAttribute class

PreserveSigAttribute.PreserveSigAttribute

System.Runtime.InteropServices.StructLayoutAttribute class

StructLayoutAttribute.StructLayoutAttribute
[StructLayoutAttribute Fields]
StructLayoutAttribute.CharSet
StructLayoutAttribute.Size

System.Runtime.InteropServices.UnmanagedType enumeration

System.Runtime.Serialization namespace

System.Runtime.Serialization.SerializationException class

SerializationException.SerializationException

System.Security namespace

System.Security.SecurityException class

SecurityException.SecurityException

System.Security.VerificationException class

VerificationException.VerificationException

System.Security.Permissions namespace

System.Security.Permissions.SecurityAction enumeration

System.Security.Policy namespace

System.Security.Policy.Evidence class

Evidence.Evidence
[Evidence Methods]
Evidence.CopyTo
Evidence.GetEnumerator
[Evidence Properties]
Evidence.Count
Evidence.IsSynchronized
Evidence.SyncRoot

System.Text namespace

System.Text.ASCIIEncoding class

ASCIIEncoding.ASCIIEncoding
[ASCIIEncoding Methods]
ASCIIEncoding.GetByteCount
ASCIIEncoding.GetBytes
ASCIIEncoding.GetCharCount
ASCIIEncoding.GetChars
ASCIIEncoding.GetMaxByteCount
ASCIIEncoding.GetMaxCharCount

System.Text.Decoder class

Decoder.Decoder
[Decoder Methods]
Decoder.Convert
Decoder.GetCharCount
Decoder.GetChars
Decoder.Reset

System.Text.Encoder class

Encoder.Encoder
[Encoder Methods]
Encoder.Convert
Encoder.GetByteCount
Encoder.GetBytes

System.Text.Encoding class

Encoding.Encoding
[Encoding Methods]
Encoding.Clone
Encoding.Convert
Encoding.Equals
Encoding.GetByteCount
Encoding.GetBytes
Encoding.GetCharCount
Encoding.GetChars
Encoding.GetDecoder
Encoding.GetEncoder
Encoding.GetEncoding
Encoding.GetHashCode
Encoding.GetMaxByteCount
Encoding.GetMaxCharCount
Encoding.GetPreamble
Encoding.GetString
[Encoding Properties]
Encoding.ASCII
Encoding.BigEndianUnicode
Encoding.CodePage
Encoding.Default
Encoding.Unicode

Encoding.UTF7
Encoding.UTF8
Encoding.WebName

System.Text.StringBuilder class

StringBuilder.StringBuilder
[StringBuilder Methods]
StringBuilder.Append
StringBuilder.AppendFormat
StringBuilder.AppendLine
StringBuilder.EnsureCapacity
StringBuilder.Equals
StringBuilder.Insert
StringBuilder.Remove
StringBuilder.Replace
StringBuilder.ToString
[StringBuilder Properties]
StringBuilder.Capacity
StringBuilder.Chars
StringBuilder.Length

System.Text.UnicodeEncoding class

UnicodeEncoding.UnicodeEncoding
[UnicodeEncoding Fields]
UnicodeEncoding.CharSize
[UnicodeEncoding Methods]
UnicodeEncoding.Equals
UnicodeEncoding.GetByteCount
UnicodeEncoding.GetBytes
UnicodeEncoding.GetCharCount
UnicodeEncoding.GetChars
UnicodeEncoding.GetDecoder
UnicodeEncoding.GetHashCode
UnicodeEncoding.GetMaxByteCount
UnicodeEncoding.GetMaxCharCount
UnicodeEncoding.GetPreamble

System.Text.UTF7Encoding class

UTF7Encoding.UTF7Encoding
[UTF7Encoding Methods]
UTF7Encoding.Equals
UTF7Encoding.GetByteCount
UTF7Encoding.GetBytes
UTF7Encoding.GetCharCount
UTF7Encoding.GetChars
UTF7Encoding.GetDecoder
UTF7Encoding.GetEncoder
UTF7Encoding.GetHashCode
UTF7Encoding.GetMaxByteCount
UTF7Encoding.GetMaxCharCount

System.Text.UTF8Encoding class

UTF8Encoding.UTF8Encoding
[UTF8Encoding Methods]
UTF8Encoding.Equals
UTF8Encoding.GetByteCount
UTF8Encoding.GetBytes
UTF8Encoding.GetCharCount
UTF8Encoding.GetChars
UTF8Encoding.GetDecoder
UTF8Encoding.GetEncoder
UTF8Encoding.GetHashCode

UTF8Encoding.GetMaxByteCount
UTF8Encoding.GetMaxCharCount
UTF8Encoding.GetPreamble

System.Text.RegularExpressions namespace

System.Text.RegularExpressions.Capture class

[Capture Methods]

Capture.ToString

[Capture Properties]

Capture.Index

Capture.Length

Capture.Value

System.Text.RegularExpressions.CaptureCollection class

[CaptureCollection Methods]

CaptureCollection.CopyTo

CaptureCollection.GetEnumerator

[CaptureCollection Properties]

CaptureCollection.Count

CaptureCollection.IsReadOnly

CaptureCollection.IsSynchronized

CaptureCollection.Item

CaptureCollection.SyncRoot

System.Text.RegularExpressions.Group class

[Group Methods]

Group.Synchronized

[Group Properties]

Group.Captures

Group.Success

System.Text.RegularExpressions.GroupCollection class

[GroupCollection Methods]

GroupCollection.CopyTo

GroupCollection.GetEnumerator

[GroupCollection Properties]

GroupCollection.Count

GroupCollection.IsReadOnly

GroupCollection.IsSynchronized

GroupCollection.Item

GroupCollection.Item

GroupCollection.SyncRoot

System.Text.RegularExpressions.Match class

[Match Methods]

Match.NextMatch

Match.Result

Match.Synchronized

[Match Properties]

Match.Empty

Match.Groups

System.Text.RegularExpressions.MatchCollection class

[MatchCollection Methods]

MatchCollection.CopyTo

MatchCollection.GetEnumerator

[MatchCollection Properties]

MatchCollection.Count

MatchCollection.IsReadOnly

MatchCollection.IsSynchronized

MatchCollection.Item
MatchCollection.SyncRoot

System.Text.RegularExpressions.MatchEvaluator delegate

System.Text.RegularExpressions.Regex class

Regex.Regex
[Regex Fields]
Regex.capnames
Regex.caps
Regex.capsize
Regex.capslist
Regex.pattern
Regex.roptions
[Regex Methods]
Regex.Escape
Regex.GetGroupNames
Regex.GetGroupNumbers
Regex.GroupNameFromNumber
Regex.GroupNumberFromName
Regex.IsMatch
Regex.Match
Regex.Matches
Regex.Replace
Regex.Split
Regex.ToString
Regex.Unescape
Regex.UseOptionC
Regex.UseOptionR
[Regex Properties]
Regex.Options
Regex.RightToLeft

System.Text.RegularExpressions.RegexOptions enumeration

System.Text.RegularExpressions.RegexRunner class

RegexRunner.RegexRunner
[RegexRunner Fields]
RegexRunner.runcrawl
RegexRunner.runcrawlpos
RegexRunner.runmatch
RegexRunner.runregex
RegexRunner.runstack
RegexRunner.runstackpos
RegexRunner.runtxt
RegexRunner.runtxtbeg
RegexRunner.runtxtend
RegexRunner.runtxtpos
RegexRunner.runtxtstart
RegexRunner.runtrack
RegexRunner.runtrackcount
RegexRunner.runtrackpos
[RegexRunner Methods]
RegexRunner.Capture
RegexRunner.CharInClass
RegexRunner.CharInSet
RegexRunner.Crawl
RegexRunner.Crawlpos
RegexRunner.DoubleCrawl
RegexRunner.DoubleStack
RegexRunner.DoubleTrack
RegexRunner.EnsureStorage

- RegexRunner.FindFirstChar
- RegexRunner.Go
- RegexRunner.InitTrackCount
- RegexRunner.IsBoundary
- RegexRunner.IsECMABoundary
- RegexRunner.IsMatched
- RegexRunner.MatchIndex
- RegexRunner.MatchLength
- RegexRunner.Popcrawl
- RegexRunner.Scan
- RegexRunner.TransferCapture
- RegexRunner.Uncapture

System.Text.RegularExpressions.RegexRunnerFactory class

- RegexRunnerFactory.RegexRunnerFactory
- [RegexRunnerFactory Methods]
- RegexRunnerFactory.CreateInstance

System.Threading namespace

System.Threading.AutoResetEvent class

- AutoResetEvent.AutoResetEvent

System.Threading.EventResetMode enumeration

System.Threading.EventWaitHandle class

- EventWaitHandle.EventWaitHandle
- [EventWaitHandle Methods]
- EventWaitHandle.Close
- EventWaitHandle.Finalize
- EventWaitHandle.Reset
- EventWaitHandle.Set
- EventWaitHandle.WaitOne

System.Threading.Interlocked class

- [Interlocked Methods]
- Interlocked.CompareExchange
- Interlocked.Decrement
- Interlocked.Exchange
- Interlocked.Increment

System.Threading.ManualResetEvent class

- ManualResetEvent.ManualResetEvent

System.Threading.Monitor class

- [Monitor Methods]
- Monitor.Enter
- Monitor.Exit
- Monitor.TryEnter

System.Threading.Mutex class

- Mutex.Mutex
- [Mutex Methods]
- Mutex.Close
- Mutex.ReleaseMutex
- Mutex.WaitOne

System.Threading.Thread class

- Thread.Thread
- [Thread Methods]
- Thread.Abort

- Thread.AllocateDataSlot
- Thread.AllocateNamedDataSlot
- Thread.FreeNamedDataSlot
- Thread.GetData
- Thread.GetDomain
- Thread.GetNamedDataSlot
- Thread.Join
- Thread.MemoryBarrier
- Thread.SetData
- Thread.SetProcessorAffinity
- Thread.Sleep
- Thread.Start
- [Thread Properties]
- Thread.CurrentThread
- Thread.IsBackground
- Thread.ManagedThreadId
- Thread.Name
- Thread.Priority

System.Threading.ThreadAbortException class

- [ThreadAbortException Properties]
- ThreadAbortException.ExceptionState

System.Threading.ThreadPool class

- [ThreadPool Methods]
- ThreadPool.GetMaxThreads
- ThreadPool.QueueUserWorkItem
- ThreadPool.SetMaxThreads

System.Threading.ThreadPriority enumeration

System.Threading.ThreadStart delegate

System.Threading.ThreadStateException class

- ThreadStateException.ThreadStateException

System.Threading.Timeout class

- [Timeout Fields]
- Timeout.Infinite

System.Threading.Timer class

- Timer.Timer
- [Timer Methods]
- Timer.Change
- Timer.Dispose

System.Threading.TimerCallback delegate

System.Threading.WaitCallback delegate

System.Threading.WaitHandle class

- WaitHandle.WaitHandle
- [WaitHandle Fields]
- WaitHandle.InvalidHandle
- [WaitHandle Methods]
- WaitHandle.Close
- WaitHandle.Dispose
- WaitHandle.WaitOne
- [WaitHandle Properties]
- WaitHandle.Handle

System.Xml namespace

System.Xml.ConformanceLevel enumeration

System.Xml.EntityHandling enumeration

System.Xml.Formatting enumeration

System.Xml.IHasXmlNode interface

[IHasXmlNode Methods]

IHasXmlNode.GetNode

System.Xml.IXmlLineInfo interface

[IXmlLineInfo Methods]

IXmlLineInfo.HasLineInfo

[IXmlLineInfo Properties]

IXmlLineInfo.LineNumber

IXmlLineInfo.LinePosition

System.Xml.IXmlNamespaceResolver interface

[IXmlNamespaceResolver Methods]

IXmlNamespaceResolver.GetNamespacesInScope

IXmlNamespaceResolver.LookupNamespace

IXmlNamespaceResolver.LookupPrefix

System.Xml.NameTable class

NameTable.NameTable

[NameTable Methods]

NameTable.Add

NameTable.Get

System.Xml.NewLineHandling enumeration

System.Xml.ReadState enumeration

System.Xml.ValidationType enumeration

System.Xml.WhitespaceHandling enumeration

System.Xml.WriteState enumeration

System.Xml.XmlAttribute class

XmlAttribute.XmlAttribute

[XmlAttribute Methods]

XmlAttribute.AppendChild

XmlAttribute.CloneNode

XmlAttribute.InsertAfter

XmlAttribute.InsertBefore

XmlAttribute.PrependChild

XmlAttribute.RemoveChild

XmlAttribute.ReplaceChild

XmlAttribute.WriteContentTo

XmlAttribute.WriteTo

[XmlAttribute Properties]

XmlAttribute.BaseURI

XmlAttribute.InnerText

XmlAttribute.InnerXml

XmlAttribute.LocalName

XmlAttribute.Name

XmlAttribute.NamespaceURI

XmlAttribute.NodeType

XmlAttribute.OwnerDocument

- XmlAttribute.OwnerElement
- XmlAttribute.ParentNode
- XmlAttribute.Prefix
- XmlAttribute.SchemaInfo
- XmlAttribute.Specified
- XmlAttribute.Value

System.Xml.XmlAttributeCollection class

- [XmlAttributeCollection Methods]
 - XmlAttributeCollection.Append
 - XmlAttributeCollection.CopyTo
 - XmlAttributeCollection.InsertAfter
 - XmlAttributeCollection.InsertBefore
 - XmlAttributeCollection.Prepend
 - XmlAttributeCollection.Remove
 - XmlAttributeCollection.RemoveAll
 - XmlAttributeCollection.RemoveAt
 - XmlAttributeCollection.SetNamedItem
- [XmlAttributeCollection Properties]
 - XmlAttributeCollection.ItemOf
 - XmlAttributeCollection.ItemOf
 - XmlAttributeCollection.ItemOf

System.Xml.XmlCDataSection class

- XmlCDataSection.XmlCDataSection
- [XmlCDataSection Methods]
 - XmlCDataSection.CloneNode
 - XmlCDataSection.WriteContentTo
 - XmlCDataSection.WriteTo
- [XmlCDataSection Properties]
 - XmlCDataSection.LocalName
 - XmlCDataSection.Name
 - XmlCDataSection.NodeType
 - XmlCDataSection.ParentNode

System.Xml.XmlCharacterData class

- XmlCharacterData.XmlCharacterData
- [XmlCharacterData Methods]
 - XmlCharacterData.AppendData
 - XmlCharacterData.DeleteData
 - XmlCharacterData.InsertData
 - XmlCharacterData.ReplaceData
 - XmlCharacterData.Substring
- [XmlCharacterData Properties]
 - XmlCharacterData.Data
 - XmlCharacterData.InnerText
 - XmlCharacterData.Length
 - XmlCharacterData.Value

System.Xml.XmlComment class

- XmlComment.XmlComment
- [XmlComment Methods]
 - XmlComment.CloneNode
 - XmlComment.WriteContentTo
 - XmlComment.WriteTo
- [XmlComment Properties]
 - XmlComment.LocalName
 - XmlComment.Name
 - XmlComment.NodeType

System.Xml.XmlConvert class

- XmlConvert.XmlConvert

[XmlConvert Methods]

XmlConvert.DecodeName
XmlConvert.EncodeLocalName
XmlConvert.EncodeName
XmlConvert.EncodeNmToken
XmlConvert.FromBinHexString
XmlConvert.ToBinHexString
XmlConvert.ToBoolean
XmlConvert.ToByte
XmlConvert.ToChar
XmlConvert.ToDateTime
XmlConvert.ToDecimal
XmlConvert.ToDouble
XmlConvert.ToGuid
XmlConvert.ToInt16
XmlConvert.ToInt32
XmlConvert.ToInt64
XmlConvert.ToSByte
XmlConvert.ToSingle
XmlConvert.ToString
XmlConvert.ToTimeSpan
XmlConvert.ToUInt16
XmlConvert.ToUInt32
XmlConvert.ToUInt64
XmlConvert.VerifyName
XmlConvert.VerifyNCName
XmlConvert.VerifyNMTOKEN
XmlConvert.VerifyTOKEN

System.Xml.XmlDateTimeSerializationMode enumeration

System.Xml.XmlDeclaration class

XmlDeclaration.XmlDeclaration
[XmlDeclaration Methods]
XmlDeclaration.CloneNode
XmlDeclaration.WriteContentTo
XmlDeclaration.WriteTo
[XmlDeclaration Properties]
XmlDeclaration.Encoding
XmlDeclaration.InnerText
XmlDeclaration.LocalName
XmlDeclaration.Name
XmlDeclaration.NodeType
XmlDeclaration.Standalone
XmlDeclaration.Value
XmlDeclaration.Version

System.Xml.XmlDocument class

XmlDocument.XmlDocument
[XmlDocument Methods]
XmlDocument.CloneNode
XmlDocument.CreateAttribute
XmlDocument.CreateCDATASection
XmlDocument.CreateComment
XmlDocument.CreateDefaultAttribute
XmlDocument.CreateDocumentFragment
XmlDocument.CreateElement
XmlDocument.CreateEntityReference
XmlDocument.CreateNode
XmlDocument.CreateProcessingInstruction
XmlDocument.CreateSignificantWhitespace
XmlDocument.CreateTextNode

- XmlDocument.CreateWhitespace
- XmlDocument.CreateXmlDeclaration
- XmlDocument.GetElementsByTagName
- XmlDocument.ImportNode
- XmlDocument.Load
- XmlDocument.LoadXml
- XmlDocument.ReadNode
- XmlDocument.Save
- XmlDocument.Validate
- XmlDocument.WriteContentTo
- XmlDocument.WriteTo

[XmlDocument Properties]

- XmlDocument.BaseURI
- XmlDocument.DocumentElement
- XmlDocument.Implementation
- XmlDocument.InnerXml
- XmlDocument.IsReadOnly
- XmlDocument.LocalName
- XmlDocument.Name
- XmlDocument.NameTable
- XmlDocument.NodeType
- XmlDocument.OwnerDocument
- XmlDocument.ParentNode
- XmlDocument.PreserveWhitespace
- XmlDocument.SchemaInfo
- XmlDocument.Schemas
- XmlDocument.XmlResolver

[XmlDocument Events]

- XmlDocument.NodeChanged
- XmlDocument.NodeChanging
- XmlDocument.NodeInserted
- XmlDocument.NodeInserting
- XmlDocument.NodeRemoved
- XmlDocument.NodeRemoving

System.Xml.XmlDocumentFragment class

XmlDocumentFragment.XmlDocumentFragment

[XmlDocumentFragment Methods]

- XmlDocumentFragment.CloneNode
- XmlDocumentFragment.WriteContentTo
- XmlDocumentFragment.WriteTo

[XmlDocumentFragment Properties]

- XmlDocumentFragment.InnerXml
- XmlDocumentFragment.LocalName
- XmlDocumentFragment.Name
- XmlDocumentFragment.NodeType
- XmlDocumentFragment.OwnerDocument
- XmlDocumentFragment.ParentNode

System.Xml.XmlElement class

XmlElement.XmlElement

[XmlElement Methods]

- XmlElement.CloneNode
- XmlElement.GetAttribute
- XmlElement.GetAttributeNode
- XmlElement.GetElementsByTagName
- XmlElement.HasAttribute
- XmlElement.RemoveAll
- XmlElement.RemoveAllAttributes
- XmlElement.RemoveAttribute
- XmlElement.RemoveAttributeAt
- XmlElement.RemoveAttributeNode

- XmlElement.SetAttribute
- XmlElement.SetAttributeNode
- XmlElement.WriteContentTo
- XmlElement.WriteTo
- [XmlElement Properties]
- XmlElement.Attributes
- XmlElement.HasAttributes
- XmlElement.InnerText
- XmlElement.InnerXml
- XmlElement.IsEmpty
- XmlElement.LocalName
- XmlElement.Name
- XmlElement.NamespaceURI
- XmlElement.NextSibling
- XmlElement.NodeType
- XmlElement.OwnerDocument
- XmlElement.ParentNode
- XmlElement.Prefix
- XmlElement.SchemaInfo

System.Xml.XmlEntityReference class

- XmlEntityReference.XmlEntityReference
- [XmlEntityReference Methods]
- XmlEntityReference.CloneNode
- XmlEntityReference.WriteContentTo
- XmlEntityReference.WriteTo
- [XmlEntityReference Properties]
- XmlEntityReference.BaseURI
- XmlEntityReference.IsReadOnly
- XmlEntityReference.LocalName
- XmlEntityReference.Name
- XmlEntityReference.NodeType
- XmlEntityReference.Value

System.Xml.XmlException class

- XmlException.XmlException
- [XmlException Properties]
- XmlException.LineNumber
- XmlException.LinePosition
- XmlException.Message
- XmlException.SourceUri

System.Xml.XmlImplementation class

- XmlImplementation.XmlImplementation
- [XmlImplementation Methods]
- XmlImplementation.CreateDocument
- XmlImplementation.HasFeature

System.Xml.XmlLinkedNode class

- [XmlLinkedNode Properties]
- XmlLinkedNode.NextSibling
- XmlLinkedNode.PreviousSibling

System.Xml.XmlNamedNodeMap class

- [XmlNamedNodeMap Methods]
- XmlNamedNodeMap.GetEnumerator
- XmlNamedNodeMap.GetNamedItem
- XmlNamedNodeMap.Item
- XmlNamedNodeMap.RemoveNamedItem
- XmlNamedNodeMap.SetNamedItem
- [XmlNamedNodeMap Properties]
- XmlNamedNodeMap.Count

System.Xml.XmlNamespaceManager class

XmlNamespaceManager.XmlNamespaceManager
[XmlNamespaceManager Methods]
XmlNamespaceManager.AddNamespace
XmlNamespaceManager.GetEnumerator
XmlNamespaceManager.GetNamespacesInScope
XmlNamespaceManager.HasNamespace
XmlNamespaceManager.LookupNamespace
XmlNamespaceManager.LookupPrefix
XmlNamespaceManager.PopScope
XmlNamespaceManager.PushScope
XmlNamespaceManager.RemoveNamespace
[XmlNamespaceManager Properties]
XmlNamespaceManager.DefaultNamespace
XmlNamespaceManager.NameTable

System.Xml.XmlNamespaceScope enumeration

System.Xml.XmlNameTable class

XmlNameTable.XmlNameTable
[XmlNameTable Methods]
XmlNameTable.Add
XmlNameTable.Get

System.Xml.XmlNode class

[XmlNode Methods]
XmlNode.AppendChild
XmlNode.Clone
XmlNode.CloneNode
XmlNode.GetEnumerator
XmlNode.GetNamespaceOfPrefix
XmlNode.GetPrefixOfNamespace
XmlNode.InsertAfter
XmlNode.InsertBefore
XmlNode.Normalize
XmlNode.PrependChild
XmlNode.RemoveAll
XmlNode.RemoveChild
XmlNode.ReplaceChild
XmlNode.SelectNodes
XmlNode.SelectSingleNode
XmlNode.Supports
XmlNode.WriteContentTo
XmlNode.WriteTo
[XmlNode Properties]
XmlNode.Attributes
XmlNode.BaseURI
XmlNode.ChildNodes
XmlNode.FirstChild
XmlNode.HasChildNodes
XmlNode.InnerText
XmlNode.InnerXml
XmlNode.IsReadOnly
XmlNode.Item
XmlNode.Item
XmlNode.LastChild
XmlNode.LocalName
XmlNode.Name
XmlNode.NamespaceURI
XmlNode.NextSibling
XmlNode.NodeType

- XmlNode.OuterXml
- XmlNode.OwnerDocument
- XmlNode.ParentNode
- XmlNode.Prefix
- XmlNode.PreviousSibling
- XmlNode.SchemaInfo
- XmlNode.Value

System.Xml.XmlNodeChangedAction enumeration

System.Xml.XmlNodeChangedEventArgs class

- XmlNodeChangedEventArgs.XmlNodeChangedEventArgs
- [XmlNodeChangedEventArgs Properties]
- XmlNodeChangedEventArgs.Action
- XmlNodeChangedEventArgs.NewParent
- XmlNodeChangedEventArgs.NewValue
- XmlNodeChangedEventArgs.Node
- XmlNodeChangedEventArgs.OldParent
- XmlNodeChangedEventArgs.OldValue

System.Xml.XmlNodeChangedEventHandler delegate

System.Xml.XmlNodeList class

- XmlNodeList.XmlNodeList
- [XmlNodeList Methods]
- XmlNodeList.GetEnumerator
- XmlNodeList.Item
- [XmlNodeList Properties]
- XmlNodeList.Count
- XmlNodeList.ItemOf

System.Xml.XmlNodeOrder enumeration

System.Xml.XmlNodeReader class

- XmlNodeReader.XmlNodeReader
- [XmlNodeReader Methods]
- XmlNodeReader.Close
- XmlNodeReader.GetAttribute
- XmlNodeReader.LookupNamespace
- XmlNodeReader.MoveToAttribute
- XmlNodeReader.MoveToElement
- XmlNodeReader.MoveToFirstAttribute
- XmlNodeReader.MoveToNextAttribute
- XmlNodeReader.Read
- XmlNodeReader.ReadAttributeValue
- XmlNodeReader.ReadContentAsBase64
- XmlNodeReader.ReadContentAsBinHex
- XmlNodeReader.ReadElementContentAsBase64
- XmlNodeReader.ReadElementContentAsBinHex
- XmlNodeReader.ReadString
- XmlNodeReader.ResolveEntity
- XmlNodeReader.Skip
- [XmlNodeReader Properties]
- XmlNodeReader.AttributeCount
- XmlNodeReader.BaseURI
- XmlNodeReader.CanReadBinaryContent
- XmlNodeReader.Depth
- XmlNodeReader.EOF
- XmlNodeReader.HasAttributes
- XmlNodeReader.HasValue
- XmlNodeReader.IsDefault
- XmlNodeReader.IsEmptyElement

- XmlNodeReader.LocalName
- XmlNodeReader.Name
- XmlNodeReader.NamespaceURI
- XmlNodeReader.NameTable
- XmlNodeReader.NodeType
- XmlNodeReader.Prefix
- XmlNodeReader.ReadState
- XmlNodeReader.SchemaInfo
- XmlNodeReader.Value
- XmlNodeReader.XmlLang
- XmlNodeReader.XmlSpace

System.Xml.XmlNodeType enumeration

System.Xml.XmlOutputMethod enumeration

System.Xml.XmlParserContext class

- XmlParserContext.XmlParserContext
- [XmlParserContext Properties]
- XmlParserContext.BaseURI
- XmlParserContext.DocTypeName
- XmlParserContext.Encoding
- XmlParserContext.InternalSubset
- XmlParserContext.NamespaceManager
- XmlParserContext.NameTable
- XmlParserContext.PublicId
- XmlParserContext.SystemId
- XmlParserContext.XmlLang
- XmlParserContext.XmlSpace

System.Xml.XmlProcessingInstruction class

- XmlProcessingInstruction.XmlProcessingInstruction
- [XmlProcessingInstruction Methods]
- XmlProcessingInstruction.CloneNode
- XmlProcessingInstruction.WriteContentTo
- XmlProcessingInstruction.WriteTo
- [XmlProcessingInstruction Properties]
- XmlProcessingInstruction.Data
- XmlProcessingInstruction.InnerText
- XmlProcessingInstruction.LocalName
- XmlProcessingInstruction.Name
- XmlProcessingInstruction.NodeType
- XmlProcessingInstruction.Target
- XmlProcessingInstruction.Value

System.Xml.XmlQualifiedName class

- XmlQualifiedName.XmlQualifiedName
- [XmlQualifiedName Fields]
- XmlQualifiedName.Empty
- [XmlQualifiedName Methods]
- XmlQualifiedName.Equals
- XmlQualifiedName.GetHashCode
- XmlQualifiedName.op_Equality
- XmlQualifiedName.op_Inequality
- XmlQualifiedName.ToString
- [XmlQualifiedName Properties]
- XmlQualifiedName.IsEmpty
- XmlQualifiedName.Name
- XmlQualifiedName.Namespace

System.Xml.XmlReader class

- XmlReader.XmlReader

[XmlReader Methods]

XmlReader.Close
XmlReader.Create
XmlReader.Dispose
XmlReader.GetAttribute
XmlReader.IsName
XmlReader.IsNameToken
XmlReader.IsStartElement
XmlReader.LookupNamespace
XmlReader.MoveToAttribute
XmlReader.MoveToContent
XmlReader.MoveToElement
XmlReader.MoveToFirstAttribute
XmlReader.MoveToNextAttribute
XmlReader.Read
XmlReader.ReadAttributeValue
XmlReader.ReadContentAs
XmlReader.ReadContentAsBase64
XmlReader.ReadContentAsBinHex
XmlReader.ReadContentAsBoolean
XmlReader.ReadContentAsDateTime
XmlReader.ReadContentAsDecimal
XmlReader.ReadContentAsDouble
XmlReader.ReadContentAsFloat
XmlReader.ReadContentAsInt
XmlReader.ReadContentAsLong
XmlReader.ReadContentAsObject
XmlReader.ReadContentAsString
XmlReader.ReadElementContentAs
XmlReader.ReadElementContentAsBase64
XmlReader.ReadElementContentAsBinHex
XmlReader.ReadElementContentAsBoolean
XmlReader.ReadElementContentAsDateTime
XmlReader.ReadElementContentAsDecimal
XmlReader.ReadElementContentAsDouble
XmlReader.ReadElementContentAsFloat
XmlReader.ReadElementContentAsInt
XmlReader.ReadElementContentAsLong
XmlReader.ReadElementContentAsObject
XmlReader.ReadElementContentAsString
XmlReader.ReadElementString
XmlReader.ReadEndElement
XmlReader.ReadInnerXml
XmlReader.ReadOuterXml
XmlReader.ReadStartElement
XmlReader.ReadString
XmlReader.ReadSubtree
XmlReader.ReadToDescendant
XmlReader.ReadToFollowing
XmlReader.ReadToNextSibling
XmlReader.ReadValueChunk
XmlReader.ResolveEntity
XmlReader.Skip

[XmlReader Properties]

XmlReader.AttributeCount
XmlReader.BaseURI
XmlReader.CanReadBinaryContent
XmlReader.CanReadValueChunk
XmlReader.CanResolveEntity
XmlReader.Depth
XmlReader.EOF
XmlReader.HasAttributes

XmlReader.HasValue
XmlReader.IsDefault
XmlReader.IsEmptyElement
XmlReader.Item
XmlReader.Item
XmlReader.Item
XmlReader.LocalName
XmlReader.Name
XmlReader.NamespaceURI
XmlReader.NameTable
XmlReader.NodeType
XmlReader.Prefix
XmlReader.QuoteChar
XmlReader.ReadState
XmlReader.SchemaInfo
XmlReader.Settings
XmlReader.Value
XmlReader.ValueType
XmlReader.XmlLang
XmlReader.XmlSpace

System.Xml.XmlReaderSettings class

XmlReaderSettings.XmlReaderSettings
[XmlReaderSettings Methods]
XmlReaderSettings.Clone
XmlReaderSettings.Reset
[XmlReaderSettings Properties]
XmlReaderSettings.CheckCharacters
XmlReaderSettings.CloseInput
XmlReaderSettings.ConformanceLevel
XmlReaderSettings.IgnoreComments
XmlReaderSettings.IgnoreProcessingInstructions
XmlReaderSettings.IgnoreWhitespace
XmlReaderSettings.LineNumberOffset
XmlReaderSettings.LinePositionOffset
XmlReaderSettings.NameTable
XmlReaderSettings.Schemas
XmlReaderSettings.ValidationFlags
XmlReaderSettings.ValidationType
XmlReaderSettings.XmlResolver
[XmlReaderSettings Events]
XmlReaderSettings.ValidationEventHandler

System.Xml.XmlResolver class

XmlResolver.XmlResolver
[XmlResolver Methods]
XmlResolver.GetEntity
XmlResolver.ResolveUri
[XmlResolver Properties]
XmlResolver.Credentials
XmlResolver.NameTable

System.Xml.XmlSignificantWhitespace class

XmlSignificantWhitespace.XmlSignificantWhitespace
[XmlSignificantWhitespace Methods]
XmlSignificantWhitespace.CloneNode
XmlSignificantWhitespace.WriteContentTo
XmlSignificantWhitespace.WriteTo
[XmlSignificantWhitespace Properties]
XmlSignificantWhitespace.LocalName
XmlSignificantWhitespace.Name
XmlSignificantWhitespace.NodeType

XmlSignificantWhitespace.ParentNode
XmlSignificantWhitespace.Value

System.Xml.XmlSpace enumeration

System.Xml.XmlText class

XmlText.XmlText
[XmlText Methods]
XmlText.CloneNode
XmlText.SplitText
XmlText.WriteContentTo
XmlText.WriteTo
[XmlText Properties]
XmlText.LocalName
XmlText.Name
XmlText.NodeType
XmlText.ParentNode
XmlText.Value

System.Xml.XmlTextReader class

XmlTextReader.XmlTextReader
[XmlTextReader Methods]
XmlTextReader.Close
XmlTextReader.GetAttribute
XmlTextReader.GetNamespacesInScope
XmlTextReader.GetRemainder
XmlTextReader.HasLineInfo
XmlTextReader.LookupNamespace
XmlTextReader.MoveToAttribute
XmlTextReader.MoveToElement
XmlTextReader.MoveToFirstAttribute
XmlTextReader.MoveToNextAttribute
XmlTextReader.Read
XmlTextReader.ReadAttributeValue
XmlTextReader.ReadBase64
XmlTextReader.ReadBinHex
XmlTextReader.ReadChars
XmlTextReader.ReadContentAsBase64
XmlTextReader.ReadContentAsBinHex
XmlTextReader.ReadElementContentAsBase64
XmlTextReader.ReadElementContentAsBinHex
XmlTextReader.ReadString
XmlTextReader.ResetState
XmlTextReader.ResolveEntity
XmlTextReader.Skip
[XmlTextReader Properties]
XmlTextReader.AttributeCount
XmlTextReader.BaseURI
XmlTextReader.CanReadBinaryContent
XmlTextReader.CanReadValueChunk
XmlTextReader.CanResolveEntity
XmlTextReader.Depth
XmlTextReader.Encoding
XmlTextReader.EntityHandling
XmlTextReader.EOF
XmlTextReader.HasValue
XmlTextReader.IsDefault
XmlTextReader.IsEmptyElement
XmlTextReader.LineNumber
XmlTextReader.LinePosition
XmlTextReader.LocalName
XmlTextReader.Name

- XmlTextReader.Namespaces
- XmlTextReader.NamespaceURI
- XmlTextReader.NameTable
- XmlTextReader.NodeType
- XmlTextReader.Normalization
- XmlTextReader.Prefix
- XmlTextReader.QuoteChar
- XmlTextReader.ReadState
- XmlTextReader.Settings
- XmlTextReader.Value
- XmlTextReader.WhitespaceHandling
- XmlTextReader.XmlLang
- XmlTextReader.XmlResolver
- XmlTextReader.XmlSpace

System.Xml.XmlTextWriter class

- XmlTextWriter.XmlTextWriter
- [XmlTextWriter Methods]
- XmlTextWriter.Close
- XmlTextWriter.Flush
- XmlTextWriter.LookupPrefix
- XmlTextWriter.WriteBase64
- XmlTextWriter.WriteBinHex
- XmlTextWriter.WriteCDATA
- XmlTextWriter.WriteCharEntity
- XmlTextWriter.WriteChars
- XmlTextWriter.WriteComment
- XmlTextWriter.WriteDocType
- XmlTextWriter.WriteEndAttribute
- XmlTextWriter.WriteEndDocument
- XmlTextWriter.WriteEndElement
- XmlTextWriter.WriteEntityRef
- XmlTextWriter.WriteFullEndElement
- XmlTextWriter.WriteName
- XmlTextWriter.WriteNmToken
- XmlTextWriter.WriteProcessingInstruction
- XmlTextWriter.WriteQualifiedName
- XmlTextWriter.WriteRaw
- XmlTextWriter.WriteStartAttribute
- XmlTextWriter.WriteStartDocument
- XmlTextWriter.WriteStartElement
- XmlTextWriter.WriteString
- XmlTextWriter.WriteSurrogateCharEntity
- XmlTextWriter.WriteWhitespace
- [XmlTextWriter Properties]
- XmlTextWriter.BaseStream
- XmlTextWriter.Formatting
- XmlTextWriter.Indentation
- XmlTextWriter.IndentChar
- XmlTextWriter.Namespaces
- XmlTextWriter.QuoteChar
- XmlTextWriter.WriteState
- XmlTextWriter.XmlLang
- XmlTextWriter.XmlSpace

System.Xml.XmlTokenizedType enumeration

System.Xml.XmlUrlResolver class

- XmlUrlResolver.XmlUrlResolver
- [XmlUrlResolver Methods]
- XmlUrlResolver.GetEntity
- XmlUrlResolver.ResolveUri

[XmlUrlResolver Properties]
XmlUrlResolver.Credentials

System.Xml.XmlWhitespace class

XmlWhitespace.XmlWhitespace
[XmlWhitespace Methods]
XmlWhitespace.CloneNode
XmlWhitespace.WriteContentTo
XmlWhitespace.WriteTo
[XmlWhitespace Properties]
XmlWhitespace.LocalName
XmlWhitespace.Name
XmlWhitespace.NodeType
XmlWhitespace.ParentNode
XmlWhitespace.Value

System.Xml.XmlWriter class

XmlWriter.XmlWriter
[XmlWriter Methods]
XmlWriter.Close
XmlWriter.Create
XmlWriter.Dispose
XmlWriter.Flush
XmlWriter.LookupPrefix
XmlWriter.WriteAttributes
XmlWriter.WriteAttributeString
XmlWriter.WriteBase64
XmlWriter.WriteBinHex
XmlWriter.WriteCData
XmlWriter.WriteCharEntity
XmlWriter.WriteChars
XmlWriter.WriteComment
XmlWriter.WriteDocType
XmlWriter.WriteElementString
XmlWriter.WriteEndAttribute
XmlWriter.WriteEndDocument
XmlWriter.WriteEndElement
XmlWriter.WriteEntityRef
XmlWriter.WriteFullEndElement
XmlWriter.WriteName
XmlWriter.WriteNmToken
XmlWriter.WriteNode
XmlWriter.WriteProcessingInstruction
XmlWriter.WriteQualifiedName
XmlWriter.WriteRaw
XmlWriter.WriteStartAttribute
XmlWriter.WriteStartDocument
XmlWriter.WriteStartElement
XmlWriter.WriteString
XmlWriter.WriteSurrogateCharEntity
XmlWriter.WriteValue
XmlWriter.WriteWhitespace
[XmlWriter Properties]
XmlWriter.Settings
XmlWriter.WriteState
XmlWriter.XmlLang
XmlWriter.XmlSpace

System.Xml.XmlWriterSettings class

XmlWriterSettings.XmlWriterSettings
[XmlWriterSettings Methods]
XmlWriterSettings.Clone

- XmlWriterSettings.Reset
- [XmlWriterSettings Properties]
- XmlWriterSettings.CheckCharacters
- XmlWriterSettings.CloseOutput
- XmlWriterSettings.ConformanceLevel
- XmlWriterSettings.Encoding
- XmlWriterSettings.Indent
- XmlWriterSettings.IndentChars
- XmlWriterSettings.NewLineChars
- XmlWriterSettings.NewLineHandling
- XmlWriterSettings.NewLineOnAttributes
- XmlWriterSettings.OmitXmlDeclaration
- XmlWriterSettings.OutputMethod

System.Xml.Schema namespace

System.Xml.Schema.IXmlSchemaInfo interface

- [IXmlSchemaInfo Properties]
- IXmlSchemaInfo.IsDefault
- IXmlSchemaInfo.IsNil
- IXmlSchemaInfo.MemberType
- IXmlSchemaInfo.SchemaAttribute
- IXmlSchemaInfo.SchemaElement
- IXmlSchemaInfo.SchemaType
- IXmlSchemaInfo.Validity

System.Xml.Schema.ValidationEventArgs class

- [ValidationEventArgs Properties]
- ValidationEventArgs.Exception
- ValidationEventArgs.Message
- ValidationEventArgs.Severity

System.Xml.Schema.ValidationEventHandler delegate

System.Xml.Schema.XmlAtomicValue class

- [XmlAtomicValue Methods]
- XmlAtomicValue.Clone
- XmlAtomicValue.ToString
- XmlAtomicValue.ValueAs
- [XmlAtomicValue Properties]
- XmlAtomicValue.IsNode
- XmlAtomicValue.TypedValue
- XmlAtomicValue.Value
- XmlAtomicValue.ValueAsBoolean
- XmlAtomicValue.ValueAsDateTime
- XmlAtomicValue.ValueAsDouble
- XmlAtomicValue.ValueAsInt
- XmlAtomicValue.ValueAsLong
- XmlAtomicValue.ValueType
- XmlAtomicValue.XmlType

System.Xml.Schema.XmlSchema class

- XmlSchema.XmlSchema
- [XmlSchema Fields]
- XmlSchema.InstanceNamespace
- XmlSchema.Namespace
- [XmlSchema Methods]
- XmlSchema.Read
- XmlSchema.Write
- [XmlSchema Properties]
- XmlSchema.AttributeFormDefault

- XmlSchema.AttributeGroups
- XmlSchema.Attributes
- XmlSchema.BlockDefault
- XmlSchema.ElementFormDefault
- XmlSchema.Elements
- XmlSchema.FinalDefault
- XmlSchema.Groups
- XmlSchema.Id
- XmlSchema.Includes
- XmlSchema.IsCompiled
- XmlSchema.Items
- XmlSchema.Notations
- XmlSchema.SchemaTypes
- XmlSchema.TargetNamespace
- XmlSchema.UnhandledAttributes
- XmlSchema.Version

System.Xml.Schema.XmlSchemaAll class

- XmlSchemaAll.XmlSchemaAll
- [XmlSchemaAll Properties]
- XmlSchemaAll.Items

System.Xml.Schema.XmlSchemaAnnotated class

- XmlSchemaAnnotated.XmlSchemaAnnotated
- [XmlSchemaAnnotated Properties]
- XmlSchemaAnnotated.Annotation
- XmlSchemaAnnotated.Id
- XmlSchemaAnnotated.UnhandledAttributes

System.Xml.Schema.XmlSchemaAnnotation class

- XmlSchemaAnnotation.XmlSchemaAnnotation
- [XmlSchemaAnnotation Properties]
- XmlSchemaAnnotation.Id
- XmlSchemaAnnotation.Items
- XmlSchemaAnnotation.UnhandledAttributes

System.Xml.Schema.XmlSchemaAny class

- XmlSchemaAny.XmlSchemaAny
- [XmlSchemaAny Properties]
- XmlSchemaAny.Namespace
- XmlSchemaAny.ProcessContents

System.Xml.Schema.XmlSchemaAnyAttribute class

- XmlSchemaAnyAttribute.XmlSchemaAnyAttribute
- [XmlSchemaAnyAttribute Properties]
- XmlSchemaAnyAttribute.Namespace
- XmlSchemaAnyAttribute.ProcessContents

System.Xml.Schema.XmlSchemaAppInfo class

- XmlSchemaAppInfo.XmlSchemaAppInfo
- [XmlSchemaAppInfo Properties]
- XmlSchemaAppInfo.Markup
- XmlSchemaAppInfo.Source

System.Xml.Schema.XmlSchemaAttribute class

- XmlSchemaAttribute.XmlSchemaAttribute
- [XmlSchemaAttribute Properties]
- XmlSchemaAttribute.AttributeSchemaType
- XmlSchemaAttribute.AttributeType
- XmlSchemaAttribute.DefaultValue
- XmlSchemaAttribute.FixedValue
- XmlSchemaAttribute.Form

- XmlSchemaAttribute.Name
- XmlSchemaAttribute.QualifiedName
- XmlSchemaAttribute.RefName
- XmlSchemaAttribute.SchemaType
- XmlSchemaAttribute.SchemaTypeName
- XmlSchemaAttribute.Use

System.Xml.Schema.XmlSchemaAttributeGroup class

- XmlSchemaAttributeGroup.XmlSchemaAttributeGroup
- [XmlSchemaAttributeGroup Properties]
- XmlSchemaAttributeGroup.AnyAttribute
- XmlSchemaAttributeGroup.Attributes
- XmlSchemaAttributeGroup.Name
- XmlSchemaAttributeGroup.QualifiedName
- XmlSchemaAttributeGroup.RedefinedAttributeGroup

System.Xml.Schema.XmlSchemaAttributeGroupRef class

- XmlSchemaAttributeGroupRef.XmlSchemaAttributeGroupRef
- [XmlSchemaAttributeGroupRef Properties]
- XmlSchemaAttributeGroupRef.RefName

System.Xml.Schema.XmlSchemaChoice class

- XmlSchemaChoice.XmlSchemaChoice
- [XmlSchemaChoice Properties]
- XmlSchemaChoice.Items

System.Xml.Schema.XmlSchemaCompilationSettings class

- XmlSchemaCompilationSettings.XmlSchemaCompilationSettings
- [XmlSchemaCompilationSettings Properties]
- XmlSchemaCompilationSettings.EnableUpaCheck

System.Xml.Schema.XmlSchemaComplexContent class

- XmlSchemaComplexContent.XmlSchemaComplexContent
- [XmlSchemaComplexContent Properties]
- XmlSchemaComplexContent.Content
- XmlSchemaComplexContent.IsMixed

System.Xml.Schema.XmlSchemaComplexContentExtension class

- XmlSchemaComplexContentExtension.XmlSchemaComplexContentExtension
- [XmlSchemaComplexContentExtension Properties]
- XmlSchemaComplexContentExtension.AnyAttribute
- XmlSchemaComplexContentExtension.Attributes
- XmlSchemaComplexContentExtension.BaseTypeName
- XmlSchemaComplexContentExtension.Particle

System.Xml.Schema.XmlSchemaComplexContentRestriction class

- XmlSchemaComplexContentRestriction.XmlSchemaComplexContentRestriction
- [XmlSchemaComplexContentRestriction Properties]
- XmlSchemaComplexContentRestriction.AnyAttribute
- XmlSchemaComplexContentRestriction.Attributes
- XmlSchemaComplexContentRestriction.BaseTypeName
- XmlSchemaComplexContentRestriction.Particle

System.Xml.Schema.XmlSchemaComplexType class

- XmlSchemaComplexType.XmlSchemaComplexType
- [XmlSchemaComplexType Properties]
- XmlSchemaComplexType.AnyAttribute
- XmlSchemaComplexType.Attributes
- XmlSchemaComplexType.AttributeUses
- XmlSchemaComplexType.AttributeWildcard
- XmlSchemaComplexType.Block
- XmlSchemaComplexType.BlockResolved

- XmlSchemaComplexType.ContentModel
- XmlSchemaComplexType.ContentType
- XmlSchemaComplexType.ContentTypeParticle
- XmlSchemaComplexType.IsAbstract
- XmlSchemaComplexType.IsMixed
- XmlSchemaComplexType.Particle

System.Xml.Schema.XmlSchemaContent class
XmlSchemaContent.XmlSchemaContent

System.Xml.Schema.XmlSchemaContentModel class
XmlSchemaContentModel.XmlSchemaContentModel
[XmlSchemaContentModel Properties]
XmlSchemaContentModel.Content

System.Xml.Schema.XmlSchemaContentProcessing enumeration

System.Xml.Schema.XmlSchemaContentType enumeration

System.Xml.Schema.XmlSchemaDatatype class
XmlSchemaDatatype.XmlSchemaDatatype
[XmlSchemaDatatype Methods]
XmlSchemaDatatype.ChangeType
XmlSchemaDatatype.IsDerivedFrom
XmlSchemaDatatype.ParseValue
[XmlSchemaDatatype Properties]
XmlSchemaDatatype.TokenizedType
XmlSchemaDatatype.TypeCode
XmlSchemaDatatype.ValueType
XmlSchemaDatatype.Variety

System.Xml.Schema.XmlSchemaDatatypeVariety enumeration

System.Xml.Schema.XmlSchemaDerivationMethod enumeration

System.Xml.Schema.XmlSchemaDocumentation class
XmlSchemaDocumentation.XmlSchemaDocumentation
[XmlSchemaDocumentation Properties]
XmlSchemaDocumentation.Language
XmlSchemaDocumentation.Markup
XmlSchemaDocumentation.Source

System.Xml.Schema.XmlSchemaElement class
XmlSchemaElement.XmlSchemaElement
[XmlSchemaElement Properties]
XmlSchemaElement.Block
XmlSchemaElement.BlockResolved
XmlSchemaElement.Constraints
XmlSchemaElement.DefaultValue
XmlSchemaElement.ElementSchemaType
XmlSchemaElement.ElementType
XmlSchemaElement.Final
XmlSchemaElement.FinalResolved
XmlSchemaElement.FixedValue
XmlSchemaElement.Form
XmlSchemaElement.IsAbstract
XmlSchemaElement.IsNillable
XmlSchemaElement.Name
XmlSchemaElement.QualifiedName
XmlSchemaElement.RefName
XmlSchemaElement.SchemaType
XmlSchemaElement.SchemaTypeName

XmlSchemaElement.SubstitutionGroup

System.Xml.Schema.XmlSchemaEnumerationFacet class
XmlSchemaEnumerationFacet.XmlSchemaEnumerationFacet

System.Xml.Schema.XmlSchemaException class
XmlSchemaException.XmlSchemaException
[XmlSchemaException Properties]
XmlSchemaException.LineNumber
XmlSchemaException.LinePosition
XmlSchemaException.SourceSchemaObject
XmlSchemaException.SourceUri

System.Xml.Schema.XmlSchemaExternal class
XmlSchemaExternal.XmlSchemaExternal
[XmlSchemaExternal Properties]
XmlSchemaExternal.Id
XmlSchemaExternal.Schema
XmlSchemaExternal.SchemaLocation
XmlSchemaExternal.UnhandledAttributes

System.Xml.Schema.XmlSchemaFacet class
XmlSchemaFacet.XmlSchemaFacet
[XmlSchemaFacet Properties]
XmlSchemaFacet.IsFixed
XmlSchemaFacet.Value

System.Xml.Schema.XmlSchemaForm enumeration

System.Xml.Schema.XmlSchemaFractionDigitsFacet class
XmlSchemaFractionDigitsFacet.XmlSchemaFractionDigitsFacet

System.Xml.Schema.XmlSchemaGroup class
XmlSchemaGroup.XmlSchemaGroup
[XmlSchemaGroup Properties]
XmlSchemaGroup.Name
XmlSchemaGroup.Particle
XmlSchemaGroup.QualifiedName

System.Xml.Schema.XmlSchemaGroupBase class
XmlSchemaGroupBase.XmlSchemaGroupBase
[XmlSchemaGroupBase Properties]
XmlSchemaGroupBase.Items

System.Xml.Schema.XmlSchemaGroupRef class
XmlSchemaGroupRef.XmlSchemaGroupRef
[XmlSchemaGroupRef Properties]
XmlSchemaGroupRef.Particle
XmlSchemaGroupRef.RefName

System.Xml.Schema.XmlSchemaldentityConstraint class
XmlSchemaldentityConstraint.XmlSchemaldentityConstraint
[XmlSchemaldentityConstraint Properties]
XmlSchemaldentityConstraint.Fields
XmlSchemaldentityConstraint.Name
XmlSchemaldentityConstraint.QualifiedName
XmlSchemaldentityConstraint.Selector

System.Xml.Schema.XmlSchemalImport class
XmlSchemalImport.XmlSchemalImport
[XmlSchemalImport Properties]
XmlSchemalImport.Annotation

[XmlSchemaImport.Namespace](#)

System.Xml.Schema.XmlSchemaInclude class

[XmlSchemaInclude.XmlSchemaInclude](#)
[\[XmlSchemaInclude Properties\]](#)
[XmlSchemaInclude.Annotation](#)

System.Xml.Schema.XmlSchemaInference class

[XmlSchemaInference.XmlSchemaInference](#)
[\[XmlSchemaInference Methods\]](#)
[XmlSchemaInference.InferSchema](#)
[\[XmlSchemaInference Properties\]](#)
[XmlSchemaInference.Occurrence](#)
[XmlSchemaInference.TypeInference](#)

System.Xml.Schema.XmlSchemaInference.InferenceOption enumeration

System.Xml.Schema.XmlSchemaInferenceException class

[XmlSchemaInferenceException.XmlSchemaInferenceException](#)

System.Xml.Schema.XmlSchemaInfo class

[XmlSchemaInfo.XmlSchemaInfo](#)
[\[XmlSchemaInfo Properties\]](#)
[XmlSchemaInfo.ContentType](#)
[XmlSchemaInfo.IsDefault](#)
[XmlSchemaInfo.IsNil](#)
[XmlSchemaInfo.MemberType](#)
[XmlSchemaInfo.SchemaAttribute](#)
[XmlSchemaInfo.SchemaElement](#)
[XmlSchemaInfo.SchemaType](#)
[XmlSchemaInfo.Validity](#)

System.Xml.Schema.XmlSchemaKey class

[XmlSchemaKey.XmlSchemaKey](#)

System.Xml.Schema.XmlSchemaKeyref class

[XmlSchemaKeyref.XmlSchemaKeyref](#)
[\[XmlSchemaKeyref Properties\]](#)
[XmlSchemaKeyref.Refer](#)

System.Xml.Schema.XmlSchemaLengthFacet class

[XmlSchemaLengthFacet.XmlSchemaLengthFacet](#)

System.Xml.Schema.XmlSchemaMaxExclusiveFacet class

[XmlSchemaMaxExclusiveFacet.XmlSchemaMaxExclusiveFacet](#)

System.Xml.Schema.XmlSchemaMaxInclusiveFacet class

[XmlSchemaMaxInclusiveFacet.XmlSchemaMaxInclusiveFacet](#)

System.Xml.Schema.XmlSchemaMaxLengthFacet class

[XmlSchemaMaxLengthFacet.XmlSchemaMaxLengthFacet](#)

System.Xml.Schema.XmlSchemaMinExclusiveFacet class

[XmlSchemaMinExclusiveFacet.XmlSchemaMinExclusiveFacet](#)

System.Xml.Schema.XmlSchemaMinInclusiveFacet class

[XmlSchemaMinInclusiveFacet.XmlSchemaMinInclusiveFacet](#)

System.Xml.Schema.XmlSchemaMinLengthFacet class

[XmlSchemaMinLengthFacet.XmlSchemaMinLengthFacet](#)

System.Xml.Schema.XmlSchemaNotation class

XmlSchemaNotation.XmlSchemaNotation
[XmlSchemaNotation Properties]
XmlSchemaNotation.Name
XmlSchemaNotation.Public
XmlSchemaNotation.System

System.Xml.Schema.XmlSchemaNumericFacet class
XmlSchemaNumericFacet.XmlSchemaNumericFacet

System.Xml.Schema.XmlSchemaObject class
XmlSchemaObject.XmlSchemaObject
[XmlSchemaObject Properties]
XmlSchemaObject.LineNumber
XmlSchemaObject.LinePosition
XmlSchemaObject.Namespaces
XmlSchemaObject.Parent
XmlSchemaObject.SourceUri

System.Xml.Schema.XmlSchemaObjectCollection class
XmlSchemaObjectCollection.XmlSchemaObjectCollection
[XmlSchemaObjectCollection Methods]
XmlSchemaObjectCollection.Add
XmlSchemaObjectCollection.Contains
XmlSchemaObjectCollection.CopyTo
XmlSchemaObjectCollection.GetEnumerator
XmlSchemaObjectCollection.IndexOf
XmlSchemaObjectCollection.Insert
XmlSchemaObjectCollection.OnClear
XmlSchemaObjectCollection.OnInsert
XmlSchemaObjectCollection.OnRemove
XmlSchemaObjectCollection.OnSet
XmlSchemaObjectCollection.Remove
[XmlSchemaObjectCollection Properties]
XmlSchemaObjectCollection.Item

System.Xml.Schema.XmlSchemaObjectEnumerator class
[XmlSchemaObjectEnumerator Methods]
XmlSchemaObjectEnumerator.MoveNext
XmlSchemaObjectEnumerator.Reset
[XmlSchemaObjectEnumerator Properties]
XmlSchemaObjectEnumerator.Current

System.Xml.Schema.XmlSchemaObjectTable class
[XmlSchemaObjectTable Methods]
XmlSchemaObjectTable.Contains
XmlSchemaObjectTable.GetEnumerator
[XmlSchemaObjectTable Properties]
XmlSchemaObjectTable.Count
XmlSchemaObjectTable.Item
XmlSchemaObjectTable.Names
XmlSchemaObjectTable.Values

System.Xml.Schema.XmlSchemaParticle class
XmlSchemaParticle.XmlSchemaParticle
[XmlSchemaParticle Properties]
XmlSchemaParticle.MaxOccurs
XmlSchemaParticle.MaxOccursString
XmlSchemaParticle.MinOccurs
XmlSchemaParticle.MinOccursString

System.Xml.Schema.XmlSchemaPatternFacet class
XmlSchemaPatternFacet.XmlSchemaPatternFacet

System.Xml.Schema.XmlSchemaRedefine class

XmlSchemaRedefine.XmlSchemaRedefine
[XmlSchemaRedefine Properties]
XmlSchemaRedefine.AttributeGroups
XmlSchemaRedefine.Groups
XmlSchemaRedefine.Items
XmlSchemaRedefine.SchemaTypes

System.Xml.Schema.XmlSchemaSequence class

XmlSchemaSequence.XmlSchemaSequence
[XmlSchemaSequence Properties]
XmlSchemaSequence.Items

System.Xml.Schema.XmlSchemaSet class

XmlSchemaSet.XmlSchemaSet
[XmlSchemaSet Methods]
XmlSchemaSet.Add
XmlSchemaSet.Compile
XmlSchemaSet.Contains
XmlSchemaSet.CopyTo
XmlSchemaSet.Remove
XmlSchemaSet.RemoveRecursive
XmlSchemaSet.Reprocess
XmlSchemaSet.Schemas
[XmlSchemaSet Properties]
XmlSchemaSet.CompilationSettings
XmlSchemaSet.Count
XmlSchemaSet.GlobalAttributes
XmlSchemaSet.GlobalElements
XmlSchemaSet.GlobalTypes
XmlSchemaSet.IsCompiled
XmlSchemaSet.NameTable
XmlSchemaSet.XmlResolver
[XmlSchemaSet Events]
XmlSchemaSet.ValidationEventHandler

System.Xml.Schema.XmlSchemaSimpleContent class

XmlSchemaSimpleContent.XmlSchemaSimpleContent
[XmlSchemaSimpleContent Properties]
XmlSchemaSimpleContent.Content

System.Xml.Schema.XmlSchemaSimpleContentExtension class

XmlSchemaSimpleContentExtension.XmlSchemaSimpleContentExtension
[XmlSchemaSimpleContentExtension Properties]
XmlSchemaSimpleContentExtension.AnyAttribute
XmlSchemaSimpleContentExtension.Attributes
XmlSchemaSimpleContentExtension.BaseTypeName

System.Xml.Schema.XmlSchemaSimpleContentRestriction class

XmlSchemaSimpleContentRestriction.XmlSchemaSimpleContentRestriction
[XmlSchemaSimpleContentRestriction Properties]
XmlSchemaSimpleContentRestriction.AnyAttribute
XmlSchemaSimpleContentRestriction.Attributes
XmlSchemaSimpleContentRestriction.BaseType
XmlSchemaSimpleContentRestriction.BaseTypeName
XmlSchemaSimpleContentRestriction.Facets

System.Xml.Schema.XmlSchemaSimpleType class

XmlSchemaSimpleType.XmlSchemaSimpleType
[XmlSchemaSimpleType Properties]
XmlSchemaSimpleType.Content

System.Xml.Schema.XmlSchemaSimpleTypeContent class
XmlSchemaSimpleTypeContent.XmlSchemaSimpleTypeContent

System.Xml.Schema.XmlSchemaSimpleTypeList class
XmlSchemaSimpleTypeList.XmlSchemaSimpleTypeList
[XmlSchemaSimpleTypeList Properties]
XmlSchemaSimpleTypeList.BaseItemType
XmlSchemaSimpleTypeList.ItemType
XmlSchemaSimpleTypeList.ItemTypeName

System.Xml.Schema.XmlSchemaSimpleTypeRestriction class
XmlSchemaSimpleTypeRestriction.XmlSchemaSimpleTypeRestriction
[XmlSchemaSimpleTypeRestriction Properties]
XmlSchemaSimpleTypeRestriction.BaseType
XmlSchemaSimpleTypeRestriction.BaseTypeName
XmlSchemaSimpleTypeRestriction.Facets

System.Xml.Schema.XmlSchemaSimpleTypeUnion class
XmlSchemaSimpleTypeUnion.XmlSchemaSimpleTypeUnion
[XmlSchemaSimpleTypeUnion Properties]
XmlSchemaSimpleTypeUnion.BaseMemberTypes
XmlSchemaSimpleTypeUnion.BaseTypes
XmlSchemaSimpleTypeUnion.MemberTypes

System.Xml.Schema.XmlSchemaTotalDigitsFacet class
XmlSchemaTotalDigitsFacet.XmlSchemaTotalDigitsFacet

System.Xml.Schema.XmlSchemaType class
XmlSchemaType.XmlSchemaType
[XmlSchemaType Methods]
XmlSchemaType.GetBuiltInComplexType
XmlSchemaType.GetBuiltInSimpleType
XmlSchemaType.IsDerivedFrom
[XmlSchemaType Properties]
XmlSchemaType.BaseSchemaType
XmlSchemaType.BaseXmlSchemaType
XmlSchemaType.Datatype
XmlSchemaType.DerivedBy
XmlSchemaType.Final
XmlSchemaType.FinalResolved
XmlSchemaType.IsMixed
XmlSchemaType.Name
XmlSchemaType.QualifiedName
XmlSchemaType.TypeCode

System.Xml.Schema.XmlSchemaUnique class
XmlSchemaUnique.XmlSchemaUnique

System.Xml.Schema.XmlSchemaUse enumeration

System.Xml.Schema.XmlSchemaValidationException class
XmlSchemaValidationException.XmlSchemaValidationException
[XmlSchemaValidationException Methods]
XmlSchemaValidationException.SetSourceObject
[XmlSchemaValidationException Properties]
XmlSchemaValidationException.SourceObject

System.Xml.Schema.XmlSchemaValidationFlags enumeration

System.Xml.Schema.XmlSchemaValidator class
XmlSchemaValidator.XmlSchemaValidator

[XmlSchemaValidator Methods]
XmlSchemaValidator.AddSchema
XmlSchemaValidator.EndValidation
XmlSchemaValidator.GetExpectedAttributes
XmlSchemaValidator.GetExpectedParticles
XmlSchemaValidator.GetUnspecifiedDefaultAttributes
XmlSchemaValidator.Initialize
XmlSchemaValidator.SkipToEndElement
XmlSchemaValidator.ValidateAttribute
XmlSchemaValidator.ValidateElement
XmlSchemaValidator.ValidateEndElement
XmlSchemaValidator.ValidateEndOfAttributes
XmlSchemaValidator.ValidateText
XmlSchemaValidator.ValidateWhitespace
[XmlSchemaValidator Properties]
XmlSchemaValidator.LineInfoProvider
XmlSchemaValidator.SourceUri
XmlSchemaValidator.ValidationEventSender
XmlSchemaValidator.XmlResolver
[XmlSchemaValidator Events]
XmlSchemaValidator.ValidationEventHandler

System.Xml.Schema.XmlSchemaValidity enumeration

System.Xml.Schema.XmlSchemaWhiteSpaceFacet class
XmlSchemaWhiteSpaceFacet.XmlSchemaWhiteSpaceFacet

System.Xml.Schema.XmlSchemaXPath class
XmlSchemaXPath.XmlSchemaXPath
[XmlSchemaXPath Properties]
XmlSchemaXPath.XPath

System.Xml.Schema.XmlSeverityType enumeration

System.Xml.Schema.XmlTypeCode enumeration

System.Xml.Schema.XmlValueGetter delegate

System.Xml.Serialization namespace

System.Xml.Serialization.IXmlSerializable interface
[IXmlSerializable Methods]
IXmlSerializable.GetSchema
IXmlSerializable.ReadXml
IXmlSerializable.WriteXml

System.Xml.Serialization.IXmlTextParser interface
[IXmlTextParser Properties]
IXmlTextParser.Normalized
IXmlTextParser.WhitespaceHandling

System.Xml.Serialization.SoapAttributeAttribute class
SoapAttributeAttribute.SoapAttributeAttribute
[SoapAttributeAttribute Properties]
SoapAttributeAttribute.AttributeName
SoapAttributeAttribute.DataType
SoapAttributeAttribute.Namespace

System.Xml.Serialization.SoapAttributeOverrides class
SoapAttributeOverrides.SoapAttributeOverrides
[SoapAttributeOverrides Methods]

- SoapAttributeOverrides.Add
- [SoapAttributeOverrides Properties]
- SoapAttributeOverrides.Item
- SoapAttributeOverrides.Item

System.Xml.Serialization.SoapAttributes class

- SoapAttributes.SoapAttributes
- [SoapAttributes Properties]
- SoapAttributes.SoapAttribute
- SoapAttributes.SoapDefaultValue
- SoapAttributes.SoapElement
- SoapAttributes.SoapEnum
- SoapAttributes.SoapIgnore
- SoapAttributes.SoapType

System.Xml.Serialization.SoapElementAttribute class

- SoapElementAttribute.SoapElementAttribute
- [SoapElementAttribute Properties]
- SoapElementAttribute.DataType
- SoapElementAttribute.ElementName
- SoapElementAttribute.IsNullable

System.Xml.Serialization.SoapEnumAttribute class

- SoapEnumAttribute.SoapEnumAttribute
- [SoapEnumAttribute Properties]
- SoapEnumAttribute.Name

System.Xml.Serialization.SoapIgnoreAttribute class

- SoapIgnoreAttribute.SoapIgnoreAttribute

System.Xml.Serialization.SoapIncludeAttribute class

- SoapIncludeAttribute.SoapIncludeAttribute
- [SoapIncludeAttribute Properties]
- SoapIncludeAttribute.Type

System.Xml.Serialization.SoapReflectionImporter class

- SoapReflectionImporter.SoapReflectionImporter
- [SoapReflectionImporter Methods]
- SoapReflectionImporter.ImportTypeMapping
- SoapReflectionImporter.IncludeType
- SoapReflectionImporter.IncludeTypes

System.Xml.Serialization.SoapTypeAttribute class

- SoapTypeAttribute.SoapTypeAttribute
- [SoapTypeAttribute Properties]
- SoapTypeAttribute.Namespace
- SoapTypeAttribute.TypeName

System.Xml.Serialization.UnreferencedObjectEventArgs class

- UnreferencedObjectEventArgs.UnreferencedObjectEventArgs
- [UnreferencedObjectEventArgs Properties]
- UnreferencedObjectEventArgs.UnreferencedId
- UnreferencedObjectEventArgs.UnreferencedObject

System.Xml.Serialization.UnreferencedObjectEventHandler delegate

System.Xml.Serialization.XmlAnyAttributeAttribute class

- XmlAnyAttributeAttribute.XmlAnyAttributeAttribute

System.Xml.Serialization.XmlAnyElementAttribute class

- XmlAnyElementAttribute.XmlAnyElementAttribute
- [XmlAnyElementAttribute Properties]

XmlAnyElementAttribute.Name
XmlAnyElementAttribute.Namespace
XmlAnyElementAttribute.Order

System.Xml.Serialization.XmlAnyElementAttributes class

XmlAnyElementAttributes.XmlAnyElementAttributes
[XmlAnyElementAttributes Methods]
XmlAnyElementAttributes.Add
XmlAnyElementAttributes.Contains
XmlAnyElementAttributes.CopyTo
XmlAnyElementAttributes.IndexOf
XmlAnyElementAttributes.Insert
XmlAnyElementAttributes.Remove
[XmlAnyElementAttributes Properties]
XmlAnyElementAttributes.Item

System.Xml.Serialization.XmlArrayAttribute class

XmlArrayAttribute.XmlArrayAttribute
[XmlArrayAttribute Properties]
XmlArrayAttribute.ElementName
XmlArrayAttribute.Form
XmlArrayAttribute.IsNullable
XmlArrayAttribute.Namespace
XmlArrayAttribute.Order

System.Xml.Serialization.XmlArrayItemAttribute class

XmlArrayItemAttribute.XmlArrayItemAttribute
[XmlArrayItemAttribute Properties]
XmlArrayItemAttribute.DataType
XmlArrayItemAttribute.ElementName
XmlArrayItemAttribute.Form
XmlArrayItemAttribute.IsNullable
XmlArrayItemAttribute.IsNullableSpecified
XmlArrayItemAttribute.Namespace
XmlArrayItemAttribute.NestingLevel
XmlArrayItemAttribute.Type

System.Xml.Serialization.XmlArrayItemAttributes class

XmlArrayItemAttributes.XmlArrayItemAttributes
[XmlArrayItemAttributes Methods]
XmlArrayItemAttributes.Add
XmlArrayItemAttributes.Contains
XmlArrayItemAttributes.CopyTo
XmlArrayItemAttributes.IndexOf
XmlArrayItemAttributes.Insert
XmlArrayItemAttributes.Remove
[XmlArrayItemAttributes Properties]
XmlArrayItemAttributes.Item

System.Xml.Serialization.XmlAttributeAttribute class

XmlAttributeAttribute.XmlAttributeAttribute
[XmlAttributeAttribute Properties]
XmlAttributeAttribute.AttributeName
XmlAttributeAttribute.DataType
XmlAttributeAttribute.Form
XmlAttributeAttribute.Namespace
XmlAttributeAttribute.Type

System.Xml.Serialization.XmlAttributeEventArgs class

[XmlAttributeEventArgs Properties]
XmlAttributeEventArgs.Attr
XmlAttributeEventArgs.LineNumber

XmlAttributeEventArgs.LinePosition
XmlAttributeEventArgs.ObjectBeingDeserialized

System.Xml.Serialization.XmlAttributeEventHandler delegate

System.Xml.Serialization.XmlAttributeOverrides class

XmlAttributeOverrides.XmlAttributeOverrides
[XmlAttributeOverrides Methods]
XmlAttributeOverrides.Add
[XmlAttributeOverrides Properties]
XmlAttributeOverrides.Item
XmlAttributeOverrides.Item

System.Xml.Serialization.XmlAttributes class

XmlAttributes.XmlAttributes
[XmlAttributes Properties]
XmlAttributes.XmlAnyAttribute
XmlAttributes.XmlAnyElements
XmlAttributes.XmlArray
XmlAttributes.XmlArrayItems
XmlAttributes.XmlAttribute
XmlAttributes.XmlChoiceIdentifier
XmlAttributes.XmlDefaultValue
XmlAttributes.XmlElements
XmlAttributes.XmlEnum
XmlAttributes.XmlIgnore
XmlAttributes.XmlIns
XmlAttributes.XmlRoot
XmlAttributes.XmlText
XmlAttributes.XmlType

System.Xml.Serialization.XmlChoiceIdentifierAttribute class

XmlChoiceIdentifierAttribute.XmlChoiceIdentifierAttribute
[XmlChoiceIdentifierAttribute Properties]
XmlChoiceIdentifierAttribute.MemberName

System.Xml.Serialization.XmlElementAttribute class

XmlElementAttribute.XmlElementAttribute
[XmlElementAttribute Properties]
XmlElementAttribute.DataType
XmlElementAttribute.ElementName
XmlElementAttribute.Form
XmlElementAttribute.IsNullable
XmlElementAttribute.Namespace
XmlElementAttribute.Order
XmlElementAttribute.Type

System.Xml.Serialization.XmlElementAttributes class

XmlElementAttributes.XmlElementAttributes
[XmlElementAttributes Methods]
XmlElementAttributes.Add
XmlElementAttributes.Contains
XmlElementAttributes.CopyTo
XmlElementAttributes.IndexOf
XmlElementAttributes.Insert
XmlElementAttributes.Remove
[XmlElementAttributes Properties]
XmlElementAttributes.Item

System.Xml.Serialization.XmlElementEventArgs class

[XmlElementEventArgs Properties]
XmlElementEventArgs.Element

XmlElementEventArgs.LineNumber
XmlElementEventArgs.LinePosition
XmlElementEventArgs.ObjectBeingDeserialized

System.Xml.Serialization.XmlElementEventHandler delegate

System.Xml.Serialization.XmlEnumAttribute class

XmlAttribute.XmlEnumAttribute
[XmlAttribute Properties]
XmlAttribute.Name

System.Xml.Serialization.XmlIgnoreAttribute class

XmlAttribute.XmlIgnoreAttribute

System.Xml.Serialization.XmlIncludeAttribute class

XmlAttribute.XmlIncludeAttribute
[XmlAttribute Properties]
XmlAttribute.Type

System.Xml.Serialization.XmlMapping class

[XmlAttribute Properties]
XmlAttribute.ElementName
XmlAttribute.Namespace

System.Xml.Serialization.XmlNamespaceDeclarationsAttribute class

XmlAttribute.XmlNamespaceDeclarationsAttribute

System.Xml.Serialization.XmlNodeEventArgs class

[XmlAttribute Properties]
XmlNodeEventArgs.LineNumber
XmlNodeEventArgs.LinePosition
XmlNodeEventArgs.LocalName
XmlNodeEventArgs.Name
XmlNodeEventArgs.NamespaceURI
XmlNodeEventArgs.NodeType
XmlNodeEventArgs.ObjectBeingDeserialized
XmlNodeEventArgs.Text

System.Xml.Serialization.XmlNodeEventHandler delegate

System.Xml.Serialization.XmlReflectionImporter class

XmlReflectionImporter.XmlReflectionImporter
[XmlReflectionImporter Methods]
XmlReflectionImporter.ImportTypeMapping
XmlReflectionImporter.IncludeType
XmlReflectionImporter.IncludeTypes

System.Xml.Serialization.XmlRootAttribute class

XmlAttribute.XmlRootAttribute
[XmlAttribute Properties]
XmlAttribute.DataType
XmlAttribute.ElementName
XmlAttribute.IsNullable
XmlAttribute.IsNullableSpecified
XmlAttribute.Namespace

System.Xml.Serialization.XmlSchemaProviderAttribute class

XmlAttribute.XmlSchemaProviderAttribute
[XmlAttribute Properties]
XmlAttribute.MethodName

System.Xml.Serialization.XmlSerializer class

- XmlSerializer.XmlSerializer
- [XmlSerializer Methods]
 - XmlSerializer.CanDeserialize
 - XmlSerializer.Deserialize
 - XmlSerializer.FromTypes
 - XmlSerializer.Serialize
- [XmlSerializer Events]
 - XmlSerializer.UnknownAttribute
 - XmlSerializer.UnknownElement
 - XmlSerializer.UnknownNode
 - XmlSerializer.UnreferencedObject

System.Xml.Serialization.XmlSerializerNamespaces class

- XmlSerializerNamespaces.XmlSerializerNamespaces
- [XmlSerializerNamespaces Methods]
 - XmlSerializerNamespaces.Add
 - XmlSerializerNamespaces.ToArray
- [XmlSerializerNamespaces Properties]
 - XmlSerializerNamespaces.Count

System.Xml.Serialization.XmlTextAttribute class

- XmlTextAttribute.XmlTextAttribute
- [XmlTextAttribute Properties]
 - XmlTextAttribute.DataType
 - XmlTextAttribute.Type

System.Xml.Serialization.XmlTypeAttribute class

- XmlTypeAttribute.XmlTypeAttribute
- [XmlTypeAttribute Properties]
 - XmlTypeAttribute.AnonymousType
 - XmlTypeAttribute.IncludeInSchema
 - XmlTypeAttribute.Namespace
 - XmlTypeAttribute.TypeName

System.Xml.Serialization.XmlTypeMapping class

- [XmlTypeMapping Properties]
 - XmlTypeMapping.TypeFullName
 - XmlTypeMapping.TypeName

System.Xml.XPath namespace

System.Xml.XPath.XPathException class

- XPathException.XPathException

System.Xml.XPath.XPathItem class

- XPathItem.XPathItem
- [XPathItem Methods]
 - XPathItem.ValueAs
- [XPathItem Properties]
 - XPathItem.IsNode
 - XPathItem.TypedValue
 - XPathItem.Value
 - XPathItem.ValueAsBoolean
 - XPathItem.ValueAsDateTime
 - XPathItem.ValueAsDouble
 - XPathItem.ValueAsInt
 - XPathItem.ValueAsLong
 - XPathItem.ValueType
 - XPathItem.XmlType

System.Xml.Xsl namespace

System.Xml.Xsl.XsltException class

XsltException.XsltException

[XsltException Properties]

XsltException.LineNumber

XsltException.LinePosition

XsltException.SourceUri

Additions to the .NET Compact Framework for Xbox 360

The .NET Compact Framework for Xbox 360 implements a subset of the .NET Compact Framework, and adds the following namespaces, types, or members.

New Methods

SetProcessorAffinity Method

Currently, the .NET Compact Framework for Xbox 360 adds just one more method. It belongs to the [System.Threading.Thread](#) class.

The [SetProcessorAffinity](#) method sets the processor or hardware thread on which a software thread will execute (the thread's *processor affinity*). The Xbox 360 has six hardware threads (two on each of three cores). Four of these can be used by applications. [SetProcessorAffinity](#) has the following signature:

C#

```
public void SetProcessorAffinity (  
    params int[] cpus  
)
```

See Also

Concepts

[Threading \(C# Programming Guide\)](#)

[Thread Pools in the .NET Compact Framework for Xbox 360](#)

Reference

[System.Threading.Thread Members](#)

[System.Threading.Thread.SetProcessorAffinity Method](#)

Thread.SetProcessorAffinity Method

Sets the processor affinity for a managed thread. Processor affinity determines the processors on which a thread runs.

Namespace: System.Threading

Assembly: mscorlib (in mscorlib.dll)

Syntax

C#

```
public void SetProcessorAffinity (
    params int[] cpus
)
```

Parameters

cpus

An array of identifiers that specify the hardware threads on which the managed thread is permitted to run.

For Xbox 360, only a single hardware thread can be specified—the *cpus* array should contain only one element. The hardware thread number must be in the range 0 through 5.

XNA Game Studio games should not use hardware threads 0 or 2; see [Remarks](#) below.

Exceptions

Exception type	Condition
InvalidOperationException	Either an invalid value was passed to the method or the thread attempting the change was different from the target thread.

Remarks

Each of the three CPU cores on the Xbox 360 has two hardware threads; the hardware thread number selects a specific hardware thread on a specific core. Hardware threads 0 and 1 are the hardware threads on core 0; 2 and 3 are the hardware threads on core 1; and 4 and 5 are the hardware threads on core 2.

Hardware Thread Number	Core
0 or 1	0
2 or 3	1
4 or 5	2

Xbox 360 software threads are associated with, and run on, only a single hardware thread at a time. The hardware thread on which a thread runs (the thread's *processor affinity*) must be set to a single hardware thread, but can be subsequently changed by calling **SetProcessorAffinity** with a different hardware thread number. When a thread is created, it is initially assigned to the current hardware thread (that is, the hardware thread on which the current thread is running).

For Xbox 360 threads, you cannot clear the thread's processor affinity by passing an empty array as the *cpus* parameter. An Xbox 360 thread will always be associated with one hardware thread.

XNA Game Studio games should not use hardware threads 0 or 2, which are reserved for the XNA Framework. The following table identifies Xbox 360 hardware thread usage.

Hardware Thread	Available
0	No, reserved for XNA Framework.
1	Yes
2	No, reserved for XNA Framework.
3	Yes
4	Yes
5	Yes

To determine the number of hardware threads supported by the hardware, use [System.Environment.ProcessorCount](#).

This method should not be used with threads created by the [ThreadPool](#) or [Timer](#) classes. It should only be called from the thread whose affinity is being changed.

See Also

Tasks

[How To: Create and Terminate Threads \(C# Programming Guide\)](#)

Concepts

[Threading \(C# Programming Guide\)](#)

[Using Threading \(C# Programming Guide\)](#)

Reference

[Thread Class](#)

[Thread Members](#)

[System.Threading Namespace](#)

[System.Environment.ProcessorCount Method](#)

[PlatformsXbox 360](#)

Thread Pools in the .NET Compact Framework for Xbox 360

A *thread pool* is a collection of threads that can be used to perform background tasks, thus freeing the primary application thread to perform other tasks asynchronously. The .NET Compact Framework for Xbox 360 supports thread pools and portions of the [System.Threading.ThreadPool](#) class. Thread pools on the Xbox 360 are slightly different than those on the .NET Framework or .NET Compact Framework.

Xbox 360 Thread Pools

The Xbox 360 CPU has six hardware threads (two on each of three cores). Each one is capable of independently executing a software thread. These hardware threads execute independently. However, the two hardware threads executing on a single core do share some resources, such as the vector unit and L1 cache.

The thread pool distributes the threads it creates evenly among four available hardware threads (two are reserved for system use). The thread pool does not try to determine the activity level of the processors before assigning threads, nor does it try to load-balance existing threads. Once a thread is assigned to a hardware thread, the thread pool will not assign that thread to a different hardware thread.

Maximum Number of Threads

The default maximum number of threads in an Xbox 360 thread pool is 256. When a new task is queued with [ThreadPool.QueueUserWorkItem](#), the thread pool first tries to assign the task to an available thread in the pool. If all threads in the thread pool are busy performing tasks, the thread pool will create a new thread to handle the task, up to the maximum number of threads set for the pool. If no threads are available and the maximum number of threads have already been created, the task is queued until a thread in the pool becomes available.

To specify the maximum number of threads for the pool, use [ThreadPool.SetMaxThreads](#). If the threads in a thread pool do not require much processing time and are cycling their tasks quickly, it may be more efficient to lower the maximum number of threads for the pool (from its default of 256). Doing so reduces the amount of context switching that occurs when a large number of ready threads are being scheduled alternately for execution on the available processors.

Using Threading

Thread pools provide an easy and efficient mechanism of scheduling asynchronous background tasks. To achieve finer control of the number, type, scheduling, and processor affinity of your application's threads, use the [System.Threading.Thread](#) class directly. You can use threads created with the [Thread](#) class instead of, or together with, thread pools.

For more information on using threading and thread pools in the .NET Framework, see [Threading](#) and [How To: Use a Thread Pool](#).

Note

The online MSDN .NET Framework documentation has been updated to show the types and members of the [System.Threading](#) namespace that are supported on the Xbox 360. Also see [Namespaces and Classes Supported by the .NET Compact Framework for Xbox 360](#) for information on determining all of the types and members supported by the .NET Compact Framework for Xbox 360.

See Also

Tasks

[How To: Use a Thread Pool \(C# Programming Guide\)](#)

Concepts

[Threading \(C# Programming Guide\)](#)

[Namespaces and Classes Supported by the .NET Compact Framework for Xbox 360](#)

Reference

[System.Threading Namespace](#)

[ThreadPool Class](#)

[ThreadPool.QueueUserWorkItem Method](#)

[ThreadPool.SetMaxThreads Method](#)

Language, Locales, and Cultures in the .NET Compact Framework for Xbox 360

Language and locale support on the Xbox 360 video game and entertainment system differs from that provided by managed culture objects in .NET. The Xbox 360 console language and locale are set by the user, and are independent of each other. For example, the console may be using the German language, but be located in Greece.

Managed culture objects (such as [CultureInfo](#)), however, are based on both language and region, where the region is dependent on the language. For example, the German language has only five valid culture regions: Austria, Germany, Liechtenstein, Luxembourg, and Switzerland.

In the .NET Compact Framework for Xbox 360, the [CultureInfo](#) returned for [CurrentCulture](#) and [CurrentUICulture](#) is based primarily on the console's current language, *not* its locale. Only if the locale matches one of a subset of regions for the current language will the locale secondarily affect the returned culture.

For example, if the console's language is German, the culture returned by [CurrentCulture](#) will be:

- 'de-AT' (German - Austria) if the console's locale is Austria,
- 'de-CH' (German - Switzerland) if the console's locale is Switzerland, and
- 'de-DE' (German - Germany) for all other locales, including locales that are valid for German (Germany, Liechtenstein, and Luxembourg) and those that are not (Greece, Korea, and India, for example).

Due to these differences, it is important to use static [CultureInfo](#) objects (rather than those returned by [CurrentCulture](#)) when displaying data from [RegionInfo](#) objects or non-Gregorian calendars. If you do not use static [CultureInfo](#) objects, note that locale mapping differences may produce unexpected or undesirable results. For example:

- If the console language is set to Traditional Chinese, and the locale is set to China, [CurrentCulture](#) and [CurrentUICulture](#) return the culture 'zh-TW' (Chinese - Taiwan).
- If the console language is set to Simplified Chinese, and the locale is set to Taiwan, [CurrentCulture](#) and [CurrentUICulture](#) return the culture 'zh-CN' (Chinese - China).

Important

Whether it is Traditional Chinese or Simplified Chinese, the console language, not the locale, determines the returned culture.

XNA developers must also be aware of locale mapping differences that can affect application development when they develop for Xbox 360 instead of Windows-based game machines. For example, the behavior of many managed functions, such as [ToUpper](#), can vary based on the value of [CurrentCulture](#) and [CurrentUICulture](#).

The following table shows the culture returned by [CurrentCulture](#) and [CurrentUICulture](#) based on the console's language and locale setting. A default culture is returned when the console's locale doesn't match a region supported for the console's language.

Console Language	Console Locale	Culture
Chinese (Simplified)	China	zh-CN (Chinese - China)
	Singapore	zh-SG (Chinese - Singapore)
	(all others)	zh-CN (Chinese - China)
Chinese (Traditional)	Hong Kong SAR	zh-HK (Chinese - Hong Kong SAR)
	Taiwan	zh-TW (Chinese - Taiwan)
	(all others)	zh-TW (Chinese - Taiwan)
English	Australia	en-AU (English - Australia)
	Canada	en-CA (English - Canada)
	Great Britain	en-GB (English - Great Britain)
	Ireland	en-IE (English - Ireland)
	New Zealand	en-NZ (English - New Zealand)
	South Africa	en-ZA (English - South Africa)
	United States	en-US (English - United States)
	(all others)	en-US (English - United States)

French	Belgium	fr-BE (French - Belgium)
	Canada	fr-CA (French - Canada)
	France	fr-FR (French - France)
	Switzerland	fr-CH (French - Switzerland)
	(all others)	fr-FR (French - France)
German	Austria	de-AT (German - Austria)
	Germany	de-DE (German - Germany)
	Switzerland	de-CH (German - Switzerland)
	(all others)	de-DE (German - Germany)
Italian	Italy	it-IT (Italian - Italy)
	Switzerland	it-CH (Italian - Switzerland)
	(all others)	it-IT (Italian - Italy)
Japanese	Japan	ja-JP (Japanese - Japan)
	(all others)	ja-JP (Japanese - Japan)
Korean	Korea	ko-KR (Korean - Korea)
	(all others)	ko-KR (Korean - Korea)
Polish	Poland	pl-PL (Polish - Poland)
	(all others)	pl-PL (Polish - Poland)
Portuguese	Brazil	pt-BR (Portuguese - Brazil)
	Portugal	pt-PT (Portuguese - Portugal)
	(all others)	pt-BR (Portuguese - Brazil)
Russian	Russia	ru-RU (Russian - Russia)
	(all others)	ru-RU (Russian - Russia)
Spanish	Chile	es-CL (Spanish - Chile)
	Colombia	es-CO (Spanish - Colombia)
	Mexico	es-MX (Spanish - Mexico)
	Spain	es-ES (Spanish - Spain)
	(all others)	es-ES (Spanish - Spain)

See Also

Concepts

[Performing Culture-Insensitive Case Changes](#)

[Performing Culture-Insensitive String Operations](#)

Reference

[CultureInfo](#)

[CurrentCulture](#)

[CurrentUICulture](#)

XNA Framework Remote Performance Monitor for Xbox 360

The XNA Framework Remote Performance Monitor for Xbox 360 is a performance analysis tool that provides run-time performance data for XNA Game Studio applications. You can monitor the data while you run the application. Also, you can use the Windows Performance Monitor (PerfMon.exe) to view the data.

In This Section

[How To: Monitor Performance at Run Time](#)

Describes performance monitoring with the XNA Framework Remote Performance Monitor for Xbox 360.

[Performance Counters in the XNA Framework](#)

Describes the performance counters used by the XNA Framework Remote Performance Monitor for Xbox 360.

[How To: Generate a Garbage Collector Heap Dump](#)

Explains how to generate a heap dump using the XNA Framework Remote Performance Monitor for Xbox 360.

[Heap Dump File Information](#)

Describes the information contained in heap dump files created by the XNA Framework Remote Performance Monitor for Xbox 360.

How To: Monitor Performance at Run Time

Describes performance monitoring with the XNA Framework Remote Performance Monitor for Xbox 360.

The XNA Framework Remote Performance Monitor for Xbox 360 is a performance analysis tool included with the XNA Framework. This tool provides a user interface that lets you view performance data at application run time. When you start your application from Remote Performance Monitor, the tool will read statistical data for the XNA Framework performance counters directly from the common language runtime (CLR).

You can also view XNA Framework performance counters using the Windows desktop tool, PerfMon.exe. Using Remote Performance Monitor in combination with PerfMon.exe gives you a live, graphical view of the performance counter data generated by the CLR.

With the Remote Performance Monitor you can:

- View performance counter data while you run an application.
- View performance counter data in a graphical format in PerfMon.exe while you run an application.
- View garbage collector heap dumps. For more information, see [How To: Generate a Garbage Collector Heap Dump](#).
- View .stat files.

For a list of the available performance counters, see [Performance Counters in the XNA Framework](#).

Using XNA Framework Remote Performance Monitor for Xbox 360

Note

Before you use Remote Performance Monitor, you must configure your Xbox 360 console to communicate with XNA Game Studio on your desktop computer. See [Configuring XNA Game Studio and Your Xbox 360 Console](#).

To run an application from Remote Performance Monitor

1. On the Xbox 360 console, start **XNA Game Studio Connect**, and then select **Launch**.
2. On your desktop computer, start **All Programs | Microsoft XNA Game Studio 3.1 | Tools | XNA Framework Remote Performance Monitor for Xbox 360**.
3. Click the **Launch Application** icon, which is the green arrow on the toolbar.
4. In the **Device** list, click the Xbox device to which you are connected.
5. In the **Application** text box, enter the game's name.

The name is set in **Project | Properties | Application | Assembly Information | Title**. This is the name of the game displayed in **XNA Game Studio Connect**.

6. In the **Arguments** text box, enter any command-line arguments for the application.
7. Verify that the application is not currently running on the device, and that the XNA Game Studio Connect screen is visible.
8. Click **OK**.

Remote Performance Monitor starts the application and displays performance data.

To use PerfMon.exe with Remote Performance Monitor

1. On your desktop computer, start **All Programs | Microsoft XNA Game Studio 3.1 | Tools | XNA Framework Remote Performance Monitor for Xbox 360**.

Caution

If you are using this tool under Windows Vista, you must run the tool with administrator privileges.

2. From the **Options** menu, click **Publish to Perfmon**.
3. Start an application from Remote Performance Monitor using the previous procedure.
4. Start PerfMon.exe.

This file is located in \Windows\System32.

5. In the left pane of PerfMon.exe, click **Performance Monitor** if it is not already selected.
6. Right-click in the right pane of PerfMon.exe, and then click **Add Counters**.
7. From the **Performance Object** list box, select a counter group name (or expand the group and select a single counter) that you want to graph in PerfMon.exe.

The group names correspond to the categories that appear when you view statistical data in Remote Performance Monitor. For example, the garbage collection group is called "Xbox 360 GC" in PerfMon.exe.

8. Click **Add**.

To save performance data

1. Start an application from Remote Performance Monitor.
2. Start PerfMon.exe, and then graph performance data.
3. From the Performance Monitor graph window, right-click and select **Save Settings As...**
4. Type a name for the file, and specify the folder in which to save the file.

The file will be saved with a .htm extension. Use Internet Explorer or Remote Performance Monitor to open and view the generated .htm file.

See Also

Concepts

[Performance Counters in the XNA Framework](#)

[How To: Generate a Garbage Collector Heap Dump](#)

[Configuring XNA Game Studio and Your Xbox 360 Console](#)

Performance Counters in the XNA Framework

The XNA Framework contains a set of predefined performance counters for analyzing the types of load your application has on the XNA Framework. The counters are not extensible.

You can generate performance statistics about your application at run time. See [How To: Monitor Performance at Run Time](#) for the procedure.

The performance counters are useful for understanding the impact of garbage collection and just-in-time (JIT) compilation on application performance. From these statistics, you can determine where optimizations should occur.

The statistics file lists performance counters with the following columns.

Column	Description
Total	Summation of occurrences
Last Datum	The last piece of information
N	The number of occurrences
Mean	The intermediate between other values
Min	The minimum value
Max	The maximum value

XNA Framework Counters

Performance counters fall into the following categories.

- [Loader](#)
- [Generics](#)
- [Locks and threads](#)
- [Garbage collection](#)
- [JIT compilation](#)
- [Exceptions](#)

Loader Counters

The following counters pertain to the portion of the common language runtime (CLR) that is responsible for resolving and loading components that are referenced by your application, such as assemblies and classes.

Name	Description
Total Program Run Time (ms)	The elapsed time, in milliseconds, from CLR invocation.
App Domains Created	The number of application domains created in the application's process.
App Domains Unloaded	The number of application domains that have been unloaded during the run time of the application.
Assemblies Loaded	The number of assemblies that have been loaded across all application domains for the application.
Classes Loaded	The number of classes that have been loaded across all application domains during the run time of the application.
Methods Loaded	The total number of methods loaded across all application domains during the run time of the application.

Generics Counters

The generics counters are loader counters that are specific to the generics feature of the XNA Framework. The CLR uses generics internally. Consequently, you might see values in these counters even if your application does not use generics explicitly.

Name	Description
Closed Types Loaded	The count of unique generic types that have been loaded across all application domains. A <i>closed type</i> is a generic class that has been bound with a specific data type. For example, an instance of <code>MyGeneric<Type<Int32></code> is a closed type.

Closed Types Loaded per Definition	<p>The maximum number of unique generic types created for a given definition across all application domains.</p> <p>For example, if an application instantiates <code>MyGenericType<Int32></code>, <code>MyGenericType<String></code>, and <code>MyGenericType<Single></code>, the number of closed types loaded for the <code>MyGenericType<T></code> definition is three (3).</p> <p>A large number (several hundred or more) in the Max column can indicate working-set pressure in terms of JIT-compiled code and CLR data structures.</p>
Open Types Loaded	<p>The count of open generic types created across all application domains.</p> <p>An <i>open type</i> is a generic class that has been loaded, but has not been bound with a specific data type. Open types are typically created only in reflection scenarios.</p> <p>For example, loading an instance of <code>MyGenericType<T></code> by using reflection will increment the Open Types Loaded counter. When the type has been bound, such as in <code>MyGenericType<Single></code>, it becomes a closed type and the Closed Types Loaded counter is incremented. The XNA Framework CLR uses reflection internally and can be the source of the open types. For example, Xml serialization and Web services use reflection internally.</p>
Closed Methods Loaded	<p>The count of unique generic methods that have been loaded across all application domains.</p> <p>A <i>closed method</i> is a generic method that has been bound with a specific data type. The type that contains the method may or may not be a generic type. For example, an instance of <code>MyType.ProcessData<Int32>()</code> is a closed method.</p>
Closed Methods Loaded per Definition	<p>The maximum number of unique generic methods created for a given definition across all application domains.</p> <p>This counter is similar to the Closed Types Loaded per Definition counter. Using the following instances of <code>MyType.ProcessData<Int32>()</code>, <code>MyType.ProcessData<String>()</code>, and <code>MyType.ProcessData<Single>()</code> will create three closed methods for the <code>MyType.ProcessData<T>()</code> definition.</p>
Open Methods Loaded	<p>The count of open generic methods created across all application domains. Open methods are typically created only in reflection scenarios. An <i>open method</i> is a generic method that has been loaded, but is not bound to a specific type.</p> <p>The XNA Framework runtime uses reflection internally and can be the source of the open methods.</p>

Locks and Threads Counters

The following counters pertain to threading, locks, timers, and thread pools. These counters can help identify threading-related performance issues by providing data on thread, timer, and lock usage during the run time of your application.

Name	Description
Threads in Thread Pool	<p>The number of threads currently in the thread pool.</p> <p>This counter counts the threads that are started by using the System.Threading.ThreadPool.QueueUserWorkItem method. This counter does <i>not</i> include threads that are started by using the System.Threading.Thread.Start method.</p>
Pending Timers	The number of timers currently queued.
Scheduled Timers	The number of timers that are currently running or scheduled to run.
Timers Delayed by Thread Pool Limit	The number of timers that have been delayed by the thread pool limit.
Work Items Queued	The number of work items queued to the thread pool.
Uncontested Monitor.Enter Calls	Number of calls made to the System.Threading.Monitor.Enter method that were not contested.
Contested Monitor.Enter Calls	<p>Number of calls made to Monitor.Enter that were contested.</p> <p>In multi-threaded applications, if <i>Thread1</i> is holding a lock and <i>Thread2</i> needs to access code that is guarded by the lock, the Contested Monitor.Enter Calls counter is incremented.</p>

Garbage Collection Counters

The following counters pertain to garbage collection operations.

Name	Description
Peak Bytes Allocated (native + managed)	The maximum number of bytes in use by the CLR, including both native and managed memory.
Managed Objects Allocated	The count of objects allocated by the garbage collector.
Unused Managed Objects Allocated	The count of objects allocated that were never used.
Managed Bytes Allocated	The count of bytes allocated by the garbage collector.
Unused Managed Bytes Allocated	The count of bytes allocated that were never used.
Managed String Objects Allocated	The number of managed string objects allocated by the garbage collector.
Bytes of String Objects Allocated	The count of bytes of string objects allocated by the garbage collector.
Garbage Collections (GC)	The number of times the garbage collector has run.
Bytes Collected by GC	The count of bytes collected by the garbage collector.
Managed Bytes in Use After GC	The number of bytes allocated to live objects after the last garbage collection.
Total Bytes in Use After GC	The number of bytes of memory, native and managed, in use after the last garbage collection.
GC Compactions	The number of times the garbage collector has compacted the heap.
Code Pitchings	The number of times the garbage collector has discarded JIT-compiled code.
Calls to GC.Collect	The number of times the application has called the System.GC.Collect method.
GC Latency Time (ms)	The total time, in milliseconds, that the garbage collector has taken to collect objects and compact the heap. The latency time of the garbage collector includes the time to collect managed objects that are no longer in scope and the compaction time, if necessary, of the GC Heap.
Pinned Objects	The count of pinned objects encountered while performing a garbage collection. <i>Pinned objects</i> are objects whose locations in memory cannot change. For example, a memory buffer that is being used to communicate between managed and unmanaged code is a pinned object. Pinned objects cannot be moved by the garbage collector during heap compaction.
Objects Moved by Compactor	The count of objects moved by the garbage collector during a compaction.
Objects Not Moved by Compactor	The count of the objects that were not moved by the garbage collector during a compaction. Objects might not be moved for a number of reasons. Some objects cannot be moved, but these objects are rare. Other objects are not moved because they are in a portion of the heap that does not need to be compacted.
Objects Finalized	The count of objects for which a finalizer has been run.
Boxed Value Types	The number of value types that have been boxed.

JIT Counters

The following counters pertain to JIT compiler operations.

Name	Description
Native Bytes Compiled JIT	The count of bytes of native code generated by the JIT compiler.
Methods Compiled JIT	The count of methods generated by the JIT compiler.
Bytes Pitched	The count of bytes of native code generated by the JIT compiler that has been discarded.
Methods Pitched	The count of methods generated by the JIT compiler that have been discarded. Unless your application was moved to the background during its run time, any positive value for the Methods Pitched counter indicates that the application ran under memory pressure.
Method Pitch Latency Time (ms)	The total time, in milliseconds, spent discarding methods generated by the JIT compiler.

Exceptions Counters

The following counters pertain to exceptions.

Name	Description
Exceptions Thrown	The count of managed exceptions that have been thrown. Because throwing an exception consumes significant resources, tracking the number of exceptions thrown by your application can help identify potential design issues.

See Also

Concepts

[How To: Monitor Performance at Run Time](#)

[Garbage Collection](#)

[Automatic Memory Management](#)

Reference

[System.Threading.Thread Class](#)

[System.Threading.ThreadPool.QueueUserWorkItem Method](#)

[System.Threading.Thread.Start Method](#)

[System.Threading.Monitor.Enter Method](#)

[System.GC.Collect Method](#)

How To: Generate a Garbage Collector Heap Dump

Explains how to generate a heap dump using the XNA Framework Remote Performance Monitor for Xbox 360.

Introduction

To help diagnose memory leaks related to objects that are kept alive inappropriately, the XNA Framework includes a remote heap dump utility. The heap dump utility is integrated with Remote Performance Monitor. Because of this, you can view the heap dump while you are running the monitor.

XNA Framework applications can experience memory leaks, although the garbage collector mitigates this issue by releasing objects that are no longer referenced.

The garbage collector walks the application roots to identify storage locations for objects on the managed heap that are no longer needed. The term *object root* refers to the object's associated application root, such as a global or static object pointer. The garbage collector will not release any object that is kept alive by a root, even if the application does not need the object any longer. For example, an object might be rooted by a static variable. Because static variables live for the life of the process, unless they are explicitly set to **null**, the objects they reference will be kept alive as long as the process is alive.

Viewing the heap dump provides the following information about the heap.

- Garbage collector heap statistics
- Root of each object in the heap
- Object references for each object in the heap

This topic describes how to view, save, open, and compare heap dumps.

Generating a Garbage Collector Heap Dump

To view a heap dump

1. Start Remote Performance Monitor.
2. Attach it to a process.

For more information, see [How To: Monitor Performance at Run Time](#).

3. In the **Live Counters window**, click **View GC Heap**.

This generates a heap dump in Remote Performance Monitor. The main view of the heap provides general statistics, a tree view of each object and its root, and a table that contains the number of instances of each type together with its cumulative size. By default, only types that are defined by the application are shown in the view. No .NET Compact Framework class library types are shown in the view.

4. Select the check boxes that correspond to the specific types that you want to see in the heap view.
5. Click **Refresh Tree** to update the view.

You can use this updated view to determine the root that keeps a particular object instance alive.

6. If you want to view object references for an object in the heap, either right-click the object and click **View Object References**, or on the **File** menu, click **View Object References**.

The object references appear in a new window.

To save a heap dump

- Save the current heap dump when you close the view of the current heap dump and you are prompted to save.

To open a previously saved heap dump

1. Start Remote Performance Monitor.
2. On the **File** menu, click **Open GC log**, and select a previously saved heap dump file.

To compare heap dumps

1. Start Remote Performance Monitor.

2. In the **Live Counters** window, click **View GC Heap** multiple times to generate heap dumps for comparison.

You can also open previously saved heap dumps to include in the comparison.

3. On the **View** menu, click **Compare Heap**.

This generates a table view of all heaps that are currently open. Each column represents a heap, with the oldest heap appearing first and the most recent heap appearing last. Each row in the column displays the number of instances of the specified type contained in the heap. If this number differs from the previous heap, the difference is shown in parentheses. For example, an entry of 152 (+90) indicates that the specified object had 90 more instances than it had in the previous heap.

See Also

Concepts

[How To: Monitor Performance at Run Time](#)

[Heap Dump File Information](#)

Heap Dump File Information

Introduction


Garbage collector heap dump files are text files that include information for every object present in the garbage collector heap at the time the dump is generated. The XNA Framework Remote Performance Monitor for Xbox 360 creates the heap dumps. For more information about generating heap dumps, see [How To: Generate a Garbage Collector Heap Dump](#).

Heap Dump File Records

Each line in the heap dump text file contains one record. The five types of records in the file are as follows.

- AppDomain record – identifies the application domain to which the dump file applies.
- Type record – describes a type that is present in the heap.
- Object record – describes an instance of a type that is present in the heap.
- Root record – describes a root instance.
- End AppDomain record – marks the end of the dump file.

Each record contains several elements separated by spaces. The first element in each line indicates the record type. The following example shows a section of a dump file.

 Note
This example is intended to show the format of the file, and may not be semantically accurate.

```
a 2 bubblecs.exe 763f9d2
t 2 System.RuntimeType
o 7c79f 2 64
t 1 System.NullReferenceException
o 7cb42 2 64
o 1ce004 1 18
t 3 System.OutOfMemoryException
o 1ce056 1d 118 1ce11f 1ce14c 1ce079 1ce113 1ce116 1ce267 1ce26a 1ce26d
r 22c81b 5 0
o 22c81e 1b 24
r 22c81e 5 0
o 22c823 1b 28
r 22c823 5 0
c bubblecs.exe 763fbef
```

AppDomain Record

An AppDomain record marks the beginning of a file section that pertains to a particular application domain. All the type, object, and root information for the garbage collector heap for that application domain is included in this section of the file. All AppDomain records close with a corresponding End AppDomain record.

The AppDomain record has four elements.

- Letter "a" – identifies the record as an AppDomain record.
- Version number – number used to track the file format of the dump file. For example, in the .NET Compact Framework version 2.0 Service Pack 2, this number is always "2".
- AppDomain name – name of the application domain used to generate the heap dump file. This is typically the name of the executable that ran in the application domain.
- Time stamp – time that the log data was generated. This value shows the number of milliseconds, represented as a hex value, since the device booted.

The following example shows an AppDomain record.

```
a 2 HH3.exe 763f9d2
```

Type Record

Type records identify types in the garbage collector heap. Each Type record contains three elements.

- Letter "t" – identifies the record as a Type record.

- Type ID – unique numeric identifier for the type. This identifier ties object instances to their types. (See the description of Object records in the next section.)

Note

The type ID is not guaranteed to be unique across dump files. For example, the identifier for the "CorporateContacts" type in the following example is not guaranteed to be "a1" in a different dump that is taken from the same application at a different point in time.

- Type name – fully qualified name of the type.

The following example shows a Type record.

```
t a1 com.PBSG.WATS.HH3.Biz.Entity.CorporateContacts
```

Object Record

Object records describe specific instances of types in the garbage collector heap. Object records have a variable number of elements.

The first four elements are required, and are as follows.

- Letter "o" – identifies the record as an Object record.
- Object ID – unique numeric identifier for the object.

Note

Object identifiers are not guaranteed to be unique across dump files. Because object identifiers can change in different dumps of the same heap, it is not possible to develop tools that identify trends across dump files.

- Type ID – identifier for the type of the object.
- Object size – size, in bytes, of the object instance.

In addition to the four required elements, Object records may have a variable number of the following element.

- Referenced object IDs – identifies other objects that the object references.

The Type record that corresponds to a particular Object record is not guaranteed to precede the Object record in the dump file.

The following example shows an Object record with three referenced object IDs.

```
o 1ce056 1d 118 1ce11f 1ce14c 1ce079
```

Root Record

Root records identify garbage collector roots.

Each Root record contains four required elements.

- Letter "r" – identifies the record as a root record.
- Object ID – identifier of the root object.
- Root descriptor – explains why the object instance is classified as a root.

The following table shows the possible values for the descriptor.

Val	Definition
0	The object instance is rooted internally by the .NET Compact Framework. For example, this includes object instances for application domains, assemblies, exceptions, and so forth.
1	The object instance is a local variable.
2	The object instance is on the finalizer queue waiting for its finalizer to run. After the finalizer runs, the object will be collected during the next garbage collection.
3	A garbage collector handle (that is, a GCHandle) exists that refers to the object instance.
4	The object instance is a static variable.
5	The object instance is a root that is specific to the garbage collection implementation in the .NET Compact Framework. For example, this includes interned strings and class descriptions.

- Root flag – flags that provide more information about the root.

The following table shows the possible flags.

Flag	Definition
0	Normal root. This value is applied to the root if none of the other flags apply.
1	The root is pinned. A root can be pinned for the following reasons. <ul style="list-style-type: none">• It refers to managed objects that are passed to native code through platform invoke.• It is referenced by a GCHandle that is created with a GCHandleType of Pinned.• A pointer in unsafe code points to the object.• A value type field within the object is referenced directly by a method argument.
2	A GCHandle whose GCHandleType is Weak refers to the object.
4	The object is pointed to by a pointer in unsafe code or is referenced by a value type field inside another object.

If the root descriptor is static (4), the root record will have an additional element.

- Root container – identifier of the type that contains the static variable.

The following example shows a root record with an additional root container element.

```
r b2753 4 0 13c
```

End AppDomain Record

End AppDomain records close the section that is initiated by a corresponding AppDomain record.

End AppDomain records have three elements.

- Letter "c" – identifies the record as an End AppDomain record.
- Application name – name of the application from which the dump file was generated. This name matches the name in the AppDomain record at the beginning of the file.
- Time stamp – time that the log data was completed. This value shows the number of milliseconds, represented as a hex value, since the device booted.

The following example shows an End AppDomain record.

```
c HH3.exe 763fbef
```

See Also

Tasks

[How To: Monitor Performance at Run Time](#)

[How To: Generate a Garbage Collector Heap Dump](#)

Reference

[GCHandle](#)

[GCHandleType](#)

Extended Tutorials

Describes how to integrate XNA Framework features and follow best practices for creating games.

Programming Guide

For more information about programming in the XNA Framework, see [Programming Guide](#).

Extended tutorials are available for the following games:

In This Section

[TopDownShooter](#)

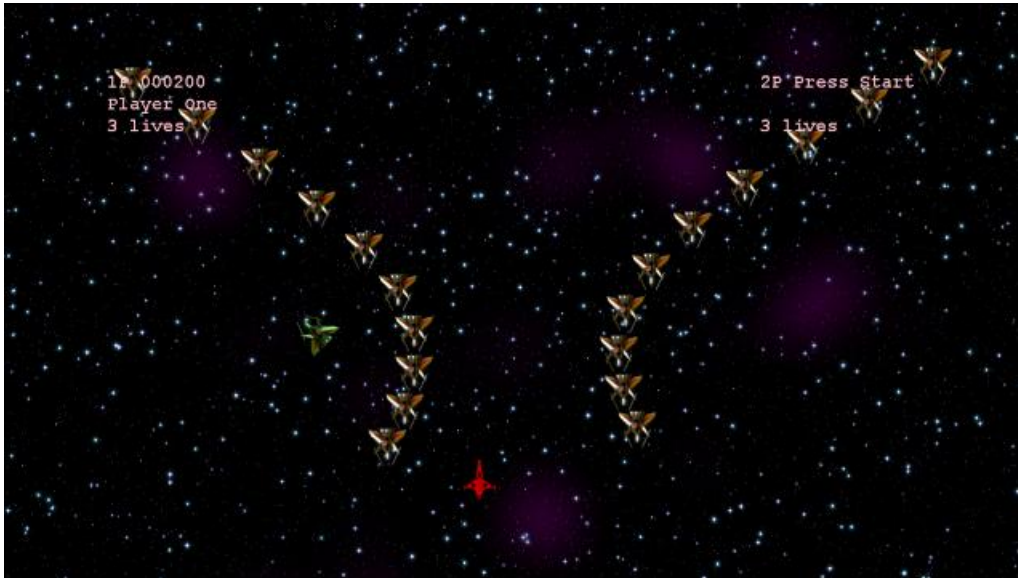
Discusses the TopDownShooter tutorial game.

[FuelCell](#)

Discusses the FuelCell 3D game tutorial.

TopDownShooter

Discusses the TopDownShooter tutorial game.



TopDownShooter is a simple 2D shooting game. In this series, we follow the development of the application from the prototype stage to a highly polished state that features multiple input configurations, more sophisticated audio and input handling, network presence, menus, and more. Along the way, we implement many of the recommended best practices for Xbox LIVE Indie Games.

In This Section

[TopDownShooter: Creating the Prototype](#)

This article is the first in a series that examines how to construct a complete game using the XNA Framework.

[TopDownShooter: Polishing the Game](#)

This article is the second in a series that examines how to construct a complete game using the XNA Framework.

TopDownShooter: Creating the Prototype

This article is the first in a series that examines how to construct a complete game using the XNA Framework.

- [The Complete Sample](#)
- [Introduction](#)
- [Game State Management](#)
- [GamePlayScreen](#)
- [GamePlayHost](#)
- [Game1](#)
- [Menu](#)
- [HelpScreen](#)
- [Miscellaneous Graphics](#)
- [Coming up Next](#)

The Complete Sample

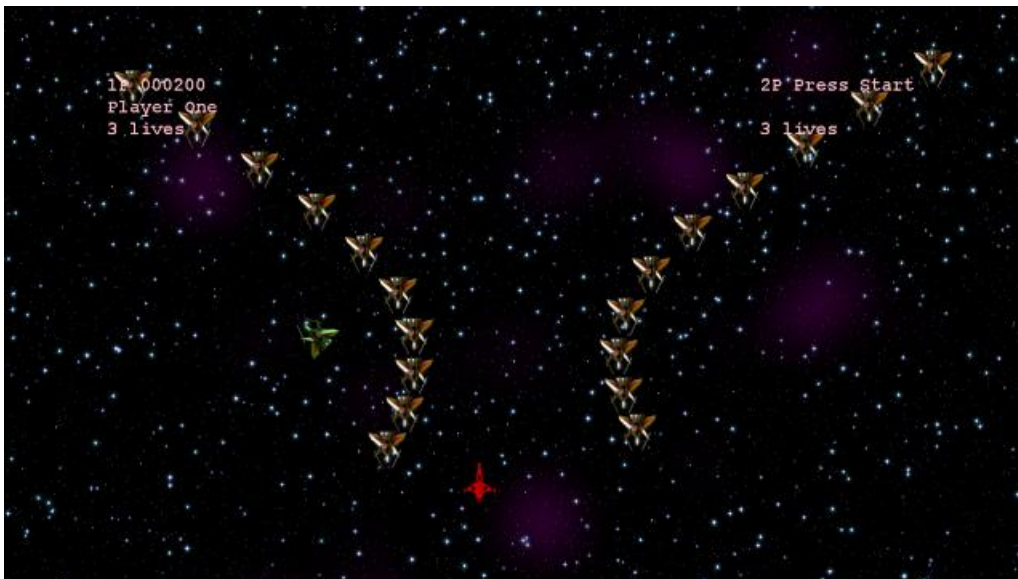
The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download TopDownShooter_1_Sample.zip.](#)

Introduction

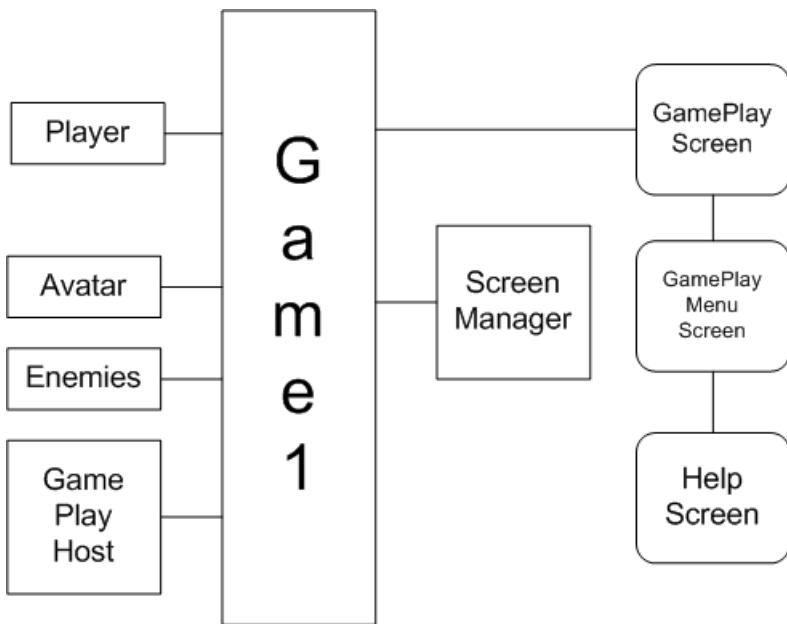
This article is the first in a series that examines how to construct a complete game using the XNA Framework. In this series, we will touch on not just graphics, but networking, input, storage, audio, and other key topics for making a fully functional game. Along the way we will show you how to implement features that go beyond gameplay – supporting multiple controller configurations, in-game volume controls, network presence, and other best practices that will make your games appear more polished and professional.

In this game, two players are represented by red and green ships in the bottom half of the screen. Vaguely insectoid enemy ships fly down from the top of the screen, and the player gains points for shooting them. A collision between an enemy and a player will destroy them both. When a player accumulates enough points, a green insectoid ship called a "seeker" will fly directly toward the player's ship until killed. If the player kills the seeker, the player can grab the lifepod the seeker ejects to gain an extra life. Below is a screenshot from the Xbox 360 version:



This version of the game is coded as a prototype. It can help us examine gameplay issues before we do the hard work of handling sound, multiple inputs, networking, etc. There are three main screens: the `GamePlayScreen`, which handles the actual gameplay, the `GamePlayMenu` which implements a popup menu over the `GamePlayScreen`, and a `HelpScreen` available from the menu.

`Game1` hosts the `GamePlayScreen`, along with helper objects: the `ScreenManager` to manage screens, the `Player` class to manage players, the `Avatar` and `Enemies` classes to handle onscreen ships, and the `GamePlayHost` to determine hits, misses, deaths, and other important events in the game.



Game State Management

The first problem to tackle when you develop your game is how to manage game state. When you play a game, you spend a lot of time in specific screens. You might go from the main menu into a network lobby, to a battle screen, and back to the network lobby, and then perhaps to an options screen, or a help screen. These screens all use art assets, and all use system memory. Some screens may need to occupy memory at the same time. For example, if the user pauses the game to change the controller configuration, you don't want to push the game screen out of system memory. Even though these screens share data, they can all be written as separate classes, and you can develop and test them individually.

To manage our screens, this sample uses code from the [Game State Management sample](#) available for free on the Creator's Club Online Web site. The Game State Management sample defines a base class called `GameScreen`, and a `ScreenManager` game component. The `ScreenManager` component keeps track of which screens are active, and the top-to-bottom order of screens.

Just like the `Game` class, the `GameScreen` class defines an `Initialize`, `LoadContent`, `Update`, and `Draw` method, and the `ScreenManager` calls these methods as a screen is loaded and displayed by the game. (`Initialize` is a new addition for this sample). Unlike the `Game` class, the `GameScreen` class defines a new method, `HandleInput`, that the `ScreenManager` will call on the topmost screen. `HandleInput` takes an instance of `InputState` as a parameter. `InputState` is a small helper class that tracks the `GamePadState` and `KeyboardState` from the last two frames. It contains some helper functions to find new button or key presses.

GamePlayScreen

In this sample, the first screen we use is the `GamePlayScreen`. The `GamePlayScreen` has two main responsibilities: draw the game, and process the input into discrete commands. Those commands are then passed back to the game. The `GamePlayScreen` doesn't act on the input directly. It doesn't hold any of the logic used for enemy AI, detecting collisions, moving the player, and so on. This is handled by the **Game** object itself, and the helper objects it uses. In essence, the `GamePlayScreen` acts like a network client, sending messages out and drawing the screen based on the current data. This is very intentional, because later on it just might be a network client. Structuring the relationship between the `GamePlayScreen` and the **Game1** object in this way gives us a preview of what our network traffic will look like when we enable network play later on.

How does `GamePlayScreen` process input? Let's take a look.

Handling Player Input

From `GamePlayScreen.cs`:

```

public override void HandleInput(InputState input)
{
    playerOneInput = ProcessPlayer(game.Main, input);

    if (game.Guest.IsPlaying)
        playerTwoInput = ProcessPlayer(game.Guest, input);

    base.HandleInput(input);
}
  
```

```

}

private static NeutralInput ProcessPlayer(Player player,
    InputState input)
{
    NeutralInput state;
    state.Fire = false;
    Vector2 stick = Vector2.Zero;
    GamePadState gpState =
        input.CurrentGamePadStates[(int)player.Controller];

    // Get gamepad state
    stick = gpState.ThumbSticks.Left;
    state.Fire = (gpState.Triggers.Right > 0);

    state.StickMovement = stick;
    return state;
}

```

The goal of `HandleInput` is to fill out a `NeutralInput` structure for each player. There are a lot of different controller configurations that could move the ship – two thumbsticks, a dpad, keyboard arrow keys, a mouse, and so on. There are also a lot of buttons on the controller or keyboard you might use to fire the ship's guns. `NeutralInput` is a structure – defined just for this game – that records the result of the input, without knowing which keys and sticks were used. Once we have the `NeutralInput` structures filled out, we leave `HandleInput`. Later on during update, this input structure is examined and real commands are issued (how far did the ship move? Did the ship fire?).

From `GamePlayScreen.cs`:

```

public void ProcessInput(int player, float totalGameSeconds,
    float elapsedGameSeconds, NeutralInput input)
{
    if (input.StickMovement.X > 0)
        accumMove.X += Move(input.StickMovement.X, elapsedGameSeconds);
    if (input.StickMovement.X < 0)
        accumMove.X -= Move(-input.StickMovement.X, elapsedGameSeconds);
    if (input.StickMovement.Y > 0)
        accumMove.Y -= Move(input.StickMovement.Y, elapsedGameSeconds);
    if (input.StickMovement.Y < 0)
        accumMove.Y += Move(-input.StickMovement.Y, elapsedGameSeconds);

    TryMove(player);

    if (input.Fire == true)
    {
        TryFire((byte)player, totalGameSeconds);
    }
}

```

Processing the input is also pretty straightforward. We gather the movement for each cardinal direction, and add it to an accumulator value. `TryMove` will move the ship when the accumulated movement is greater than half a pixel. This keeps us from sending movement commands for movements that won't actually change the ship's position on screen. This will save you a little network bandwidth later. `TryFire` lets the player fire when enough time has passed between shots.

From `GamePlayScreen.cs`:

```

private void TryMove(int player)
{
    if (accumMove.Length() > .5)
    {
        // Find out how far we are allowed to move
        Vector2 move = game.ships[player].VerifyMove(accumMove);

        // Send our movement event to Game1
    }
}

```

```

        Vector2 pos = game.ships[player].position + move;
        game.ShipMove((byte)player, pos);

        // Zero out accumulated input
        accumMove = Vector2.Zero;
    }
}

private void TryFire(byte player, float totalGameSeconds)
{
    if (game.ships[player].status != ObjectStatus.Active)
        return;

    // if the time between the last shot and the current
    // time is less than 1/ROF, don't fire
    if (game.ships[player].VerifyFire(totalGameSeconds))
    {
        game.ShipFire(player, totalGameSeconds);
    }
}

```

`TryMove` notifies the **Game1** object when the ship has moved by calling the `ShipMove` method. `TryFire` calls `ShipFire` in the same way. These are our notifications from the client to the "server" that the user has done something important. But it's up to `Game1` to decide whether to update the ship.

Drawing the game

The other major responsibility of `GamePlayScreen` is to draw the game in progress. This means drawing good guys, bad guys, bullets, explosions, and a scoreboard. To do that, it needs access to the position data for the player ship, enemy ships, and bullets. This data is stored in the **Game1** object, in an **Enemies** object and two **Ship** objects (one for each player). It also needs to load copies of the sprite graphics. This is handled in `LoadContent`:

From `GamePlayScreen.cs`:

```

Texture2D bullet;
Texture2D ship;
Vector2 bulletorigin;
Vector2 shiporigin;
Texture2D villain;
Vector2 villainOrigin;
Texture2D gameover;
Vector2 gameoverOrigin;
SpriteFont arcade;

public override void LoadContent()
{
    bullet = game.Content.Load<Texture2D>("Photon");
    ship = game.Content.Load<Texture2D>("Playership");
    shiporigin = new Vector2(ship.Width / 2, ship.Height / 2);
    bulletorigin = new Vector2(bullet.Width / 2, bullet.Height / 2);

    villain = game.Content.Load<Texture2D>("Villain");
    villainOrigin = new Vector2(villain.Width / 2, villain.Height / 2);

    gameover = game.Content.Load<Texture2D>("gameover");
    gameoverOrigin =
        new Vector2(gameover.Width / 2, gameover.Height / 2);

    arcade = game.Content.Load<SpriteFont>("Arcade");
    Explosion.Texture = game.Content.Load<Texture2D>("Explosion");

    base.LoadContent();
}

```

For convenience, we separate each part of the drawing process into separate methods. Note that the raw data for drawing comes from the `Game1` class.

From `GamePlayScreen.cs`:

```
public override void Draw(GameTime gameTime)
{
    base.Draw(gameTime);

    // Draw each player
    DrawPlayer(game.ships[0], game.TotalGameSeconds);
    DrawPlayer(game.ships[1], game.TotalGameSeconds);

    // Draw the enemies
    DrawEnemies(game.TotalGameSeconds);

    // Draw each player's bullets
    DrawBullets(game.ships[0].bullets, game.TotalGameSeconds);
    DrawBullets(game.ships[1].bullets, game.TotalGameSeconds);

    // Draw the score
    DrawUI();

    // If the game is over, draw a special graphic
    DrawGameOver();
}
```

First up is our hero, the player's ship. To draw the ship, we borrow an initialized `SpriteBatch` from the `ScreenManager`, and then call `Begin`. If you're not familiar with `SpriteBatch`, it's a class that renders textures as on-screen sprites. It does this by creating quads (two-dimensional squares) in 3D space and applying the texture to the quad. Rotating the quad rotates the sprite; stretching the quad stretches the sprite, and so on. `SpriteBatch` draws the sprites all at once (hence "batch") in between `Begin` and `End` calls. `Begin` specifies what kind of rendering to use, and how to handle batches of sprites. We specify `SpriteBlendMode.AlphaBlend` to turn on alpha blending, and `SpriteSortMode.Deferred` indicates that we want to wait until `End` is called to draw all the sprites.

From `GamePlayScreen.cs`:

```
private void DrawPlayer(Avatar state, double totalGameSeconds)
{
    // Initialize our own SpriteBatch. We want default Alpha blending,
    // and the render states set by SpriteBatch to be reset
    // when End() is called.
    SpriteBatch batch = this.ScreenManager.SpriteBatch;
    batch.Begin(SpriteBlendMode.AlphaBlend, SpriteSortMode.Deferred,
        SaveStateMode.SaveState);

    switch (state.status)
    {
    case ObjectStatus.Inactive:
        break; // Draw nothing
    case ObjectStatus.Active:
        // Draws our avatar at the current position with no tinting
        batch.Draw(ship, state.position, null, state.color, 0,
            shiporigin, 1.0f, SpriteEffects.None, 0.5f);
        break;
    case ObjectStatus.Dying:
        Explosion.Draw(batch, state.position, 0, totalGameSeconds,
            state.deathTimeTotalSeconds,
            state.deathTimeTotalSeconds + Avatar.RespawnTime);
        break;
    case ObjectStatus.Immune://blink 5 times per second
        if (((int)(totalGameSeconds * 10) % 2) == 0)
            batch.Draw(ship, state.position, null, Color.Silver, 0,
                shiporigin, 1.0f, SpriteEffects.None, 0.5f);
        break;
    default:

```

```

        break;
    }
    batch.End();
}

```

With an active [SpriteBatch](#), we draw the appropriate graphic depending on what state the player is in. If the player is `Active`, we draw the ship with the normal color. If the player is `Dying`, we use the `Explosion` class to draw an explosion at the player's location. After they explode, the players become `Immune` to damage as they respawn. In this case we draw the ship in silver and blink the image as a cue that they won't be immune for very long, and they should move to a safe place.

Drawing the bullets is much the same process, with one difference. When a bullet is fired, we record the X and Y coordinate of the new bullet in a `Vector3`, and the time the bullet was fired becomes the Z coordinate. From then on, we do not update the X and Y coordinate of the bullet. Instead, we use the fact that all bullets fly at a constant speed to calculate the current location of the bullet, as in the following code:

From `GamePlayHost.cs`:

```

public static float bulletSpeed = 300;
public static Vector2 FindBulletPosition(Vector3 bullet,
    double totalGameSeconds)
{
    Vector2 pos = Vector2.Zero;
    pos.X = bullet.X;
    pos.Y =
        bullet.Y - (bulletSpeed * ((float)totalGameSeconds - bullet.Z));
    return pos;
}

```

To draw the bullets, we call `FindBulletPosition` for each bullet. If we find a bullet position that is actually offscreen, we remove the bullet from the list:

From `GamePlayScreen.cs`:

```

private void DrawBullets(List<Vector3> bullets, double TotalGameSeconds)
{
    SpriteBatch batch = this.ScreenManager.SpriteBatch;
    batch.Begin();
    Vector2 pos;
    for (int i = 0; i < bullets.Count; i++)
    {
        pos = GamePlayHost.FindBulletPosition(bullets[i],
            TotalGameSeconds);
        if (pos.Y > -10) // bullet is far off the top of the screen
        {
            batch.Draw(bullet, pos, null, Color.White, 0,
                bullet.Origin, 1.0f, SpriteEffects.None, 0.6f);
        }
        else
        {
            // Since we are iterating each bullet anyway, now is a
            // good time to remove bullets that go offscreen.
            bullets.RemoveAt(i);
        }
    }
    batch.End();
}

```

Drawing enemies is a little more complicated. Just like the bullets, we don't actually store the current position of the enemies. But the enemies don't move in a straight line – they swoop around the screen. For that reason, this sample uses the XNA

Framework [Curve](#) class to store one curve for the Y axis of the enemy, and one curve for the X axis. As time elapses, the enemy moves along each curve until eventually the enemy disappears offscreen. Then we reset the position on the curve back to the beginning and start again. So in order to draw each enemy, we need to determine the new position of each enemy. That's done with the following method of the `Enemies` class (the `EnemyInfo` class holds the proper curve for a given type of enemy):

From Enemies.cs:

```
public static Vector2 GetPosition(float position, EnemyInfo info)
{
    Vector2 pos = Vector2.Zero;
    pos.Y = info.YCurve.Evaluate(position) * info.ScreenBounds.Height;
    pos.X = info.XCurve.Evaluate(position) * info.ScreenBounds.Width;
    return pos;
}
```

Once we have the position of an enemy, we just have to draw the position (if the enemy is alive), or draw an explosion if the enemy is dying. Seekers do not use curves – they each have an X and Y position just like the players.

From GameplayScreen.cs:

```
private void DrawEnemies(float TotalGameSeconds)
{
    SpriteBatch batch = this.ScreenManager.SpriteBatch;
    batch.Begin();
    // Draw first enemy type
    foreach (EnemyState enemy in game.enemies.waves)
    {
        Vector2 pos = Enemies.GetPosition(TotalGameSeconds, enemy);
        if (enemy.status == ObjectStatus.Active)
        {
            batch.Draw(villain, pos, null, Color.White, 0,
                villainOrigin, 1.0f, SpriteEffects.None, .6f);
        }
        else if (enemy.status == ObjectStatus.Dying)
        {
            // I'm exploding!
            Explosion.Draw(batch, pos, enemy.deathTimeTotalSeconds,
                TotalGameSeconds,
                enemy.deathTimeTotalSeconds,
                enemy.deathTimeTotalSeconds + 1.5);
        }
    }
    // Draw seekers, if any
    foreach (SeekerState seeker in game.enemies.seekers)
    {
        if (seeker.status == ObjectStatus.Active)
        {
            batch.Draw(villain, seeker.position, null, Color.LightGreen,
                seeker.angle, villainOrigin, 1.0f,
                SpriteEffects.None, .6f);
        }
        else if (seeker.status == ObjectStatus.Dying)
        {
            // Draw the powerup
            batch.Draw(villain, seeker.position, null, Color.White,
                seeker.angle, villainOrigin, 0.5f,
                SpriteEffects.None, .6f);

            // Draw the explosion
            Explosion.Draw(batch, seeker.position,
                seeker.deathTimeTotalSeconds, TotalGameSeconds,
                seeker.deathTimeTotalSeconds,
                seeker.deathTimeTotalSeconds + .5);
        }
    }
    batch.End();
}
```

```
}
```

Our last major task for the `GamePlayScreen`, then, is drawing the score. First, we construct a multi-line string for each player, giving the score, name, and how many lives remain for each player. Then we use a `SpriteBatch` with the `DrawString` method (instead of the `Draw` method we've been using for sprites). `DrawString` takes a given string and draws the letters from a given `SpriteFont`. We loaded the `SpriteFont`, named `arcade`, in `LoadContent`. We use the `MeasureString` method to make sure that we will have enough room to print the second player's gamertag.

From `GamePlayScreen.cs`:

```
private void DrawUI()
{
    string P2Text = (game.ships[1].Player.IsPlaying) ? "2P " +
        game.ships[1].score.ToString("000000") :
        "2P Press Start";
    P2Text += "\n\r" + game.ships[1].Player.Name;
    P2Text += "\n\r" + game.ships[1].lives.ToString() + " lives";

    string P1Text = "1P " + game.ships[0].score.ToString("000000");

    P1Text += "\n\r" + game.ships[0].Player.Name;
    P1Text += "\n\r" + game.ships[0].lives.ToString() + " lives";

    Vector2 Pos = new Vector2(uiBounds.X, uiBounds.Y+10);

    SpriteBatch batch = this.ScreenManager.SpriteBatch;
    batch.Begin(SpriteBlendMode.AlphaBlend, SpriteSortMode.Deferred,
        SaveStateMode.SaveState);
    batch.DrawString(arcade, P1Text, Pos, Color.Pink);

    // Gamertags can be 15 characters
    int gamertagwidth =
        (int)(arcade.MeasureString("123456789012345").X);
    Pos.X = uiBounds.Right - gamertagwidth;

    batch.DrawString(arcade, P2Text, Pos, Color.Pink);
    batch.End();
}
```

GamePlayHost

Now that we can draw the game, let's take a look at the brains of the game: `GamePlayHost`. The gameplay logic is sequestered into a separate object for a couple reasons, but mainly because in a network game, only the host actually needs it. Let's begin by taking a look at the `Update` method for `GamePlayHost`:

From `GamePlayHost.cs`:

```
public void Update(GameTime gameTime)
{
    if (!bActive)
        return;

    SpawnWaves(Game.TotalGameSeconds);

    HandleCollisions(Game.ships[0], 0,
        Game.enemies, Game.TotalGameSeconds);
    HandleCollisions(Game.ships[1], 1,
        Game.enemies, Game.TotalGameSeconds);

    SpawnSeekers();

    for (int j = 0; j < Game.enemies.seekers.Length; j++)
    {
        UpdateSeeker(Game.TotalGameSeconds,
```



```

        (float)gameTime.ElapsedGameTime.TotalSeconds,
        Game.enemies.seekers[j], (byte)j, Game.ships);
    }
}

```

After checking to see if the host is active (and returning early if it's not), we farm out the major logic tasks to other methods. `SpawnWaves` checks to see if it's time to create another wave of enemies. `HandleCollisions` runs once for each player ship on screen, checking the player's position and the player's bullets against the currently active enemies. Lastly, "Seekers" are a special type of enemy that chase the player, and so `GamePlayHost` determines when they get created, and how they move.

From `GamePlayHost.cs`:

```

private void SpawnWaves(double TotalGameSeconds)
{
    // If it's time for a new wave and no enemies are
    // left from the last wave
    if ((Game.enemies.ActiveEnemies == 0) &&
        (TotalGameSeconds >= NextWaveEarliestSeconds))
    {
        // If the wave was killed in one pass, both players get a bonus
        if (bKillAllInOnePass)
        {
            //Game.state.Players[0].score += 1000;
            Game.IncreaseScore(0, 1000);

            // PlayerTwo doesn't get a bonus if they're not playing
            if (Game.ships[1].status != ObjectStatus.Inactive)
                Game.IncreaseScore(1, 1000);
        }

        // Spawn the next wave
        Game.SpawnNextWave((float) TotalGameSeconds);

        // Reset our booleans
        bLastWaveBasic = !bLastWaveBasic;
        bKillAllInOnePass = true;

        // Reset the wave timer
        NextWaveEarliestSeconds = (float)TotalGameSeconds + 2.0f;
    }
    else if (TotalGameSeconds >= NextWaveEarliestSeconds)
    {
        NextWaveEarliestSeconds = (float)TotalGameSeconds + 2.0f;
        bKillAllInOnePass = false;
    }
}

```

The most important thing to note about `SpawnWaves` is that it doesn't actually create the new enemies. `Game.SpawnNextWave` does that. `GamePlayHost.SpawnWaves` just determines if it's the proper time to do so. `Game.SpawnNextWave` creates new enemies because in this sample `Game1` owns all the data. In later versions, `Game1` will spawn new waves in response to a network message from the host.

The second thing to note is that `SpawnWaves` decides if players get a bonus for clearing the wave in one try, and uses methods on `Game` to increase their score.

In a game like this, or any game that involves shooting, you're going to be spending a lot of CPU time detecting collisions. Detecting collisions between all the objects in your world can be a laborious process. For example, if you have bullets in your game that hit objects, you have to check the position of each bullet to the location of each object a bullet might hit. As the number of objects and bullets grows, the CPU time spent detecting collisions goes up geometrically.

Fortunately, there a lot of tricks you can use. We won't detail them all here, but they all involve reducing the number of objects to be tested, or dividing them into batches. In this case, the number of objects doesn't grow too much. We know the maximum number of enemies, of players, and bullets. Therefore, we compare each player to the data set for the enemies. Comparing two sets of objects to see if any collide (in this case, bullets and enemies) usually involves two nested loops. Inside the inner loop,

you test one object against another:

From GameplayHost.cs:

```
public void HandleCollisions(Avatar player, byte playerId,
    Enemies enemies, float totalGameSeconds)
{
    // Skip a lot of processing if the player isn't playing
    if (player.status == ObjectStatus.Inactive)
        return;

    for (int j = 0; j < enemies.waves.Length; j++)
    {
        if (enemies.waves[j].status == ObjectStatus.Active)
        {
            Vector2 enemyPos = Enemies.GetPosition(totalGameSeconds,
                enemies.waves[j]);
            for (int i = 0; i < player.bullets.Count; i++)
            {
                {
                    // Did player kill the enemy?
                    if (DetectCollision(player.bullets[i],
                        enemyPos, totalGameSeconds))
                    {
                        player.bullets.RemoveAt(i);
                        Game.EnemyKilled((byte)j, (byte)playerId);
                    }
                }
            }
            // Did the player crash into the enemy?
            if (DetectCrash(player, enemyPos,
                Enemies.GetInfoForType(enemies.waves[j].type)))
            {
                Game.EnemyKilled((byte)j, (byte)playerId);
                DestroyPlayer((byte)playerId);
            }
        }
    }
}
```

If the player is playing, we start with the list of enemies, determining the position of each one, then checking each of our player bullets to see if this enemy was hit. We then to check to make sure that the player didn't crash into the enemy, either. If a bullet hits an enemy, we inform the **Game1** object of the results of the collision. If a player hits the enemy, we inform the **Game1** object of the (slightly different) results.

The second half of `HandleCollisions` repeats this process for each Seeker type enemy onscreen (if any):

From GameplayHost.cs:

```
for (int k = 0; k < enemies.seekers.Length; k++)
{
    if (enemies.seekers[k].status == ObjectStatus.Active)
    {
        for (int l = 0; l < player.bullets.Count; l++)
        {
            // Did player kill seeker?
            if (DetectCollision(player.bullets[l],
                enemies.seekers[k].position, totalGameSeconds))
            {
                Game.SeekerDestroyed((byte)k);
                player.bullets.RemoveAt(l);
            }
        }
    }
    // Did the player crash into the seeker?
    if (DetectCrash(player, enemies.seekers[k].position,
        Enemies.GetInfoForType(EnemyType.Seeker)))
```

```

    {
        // The seeker is live, crash the player
        if (enemies.seekers[k].status == ObjectStatus.Active)
        {
            Game.SeekerCrashed((byte)k);
            DestroyPlayer((byte)playerId);
        }
        // The seeker is a popup, reward the player
        else if (enemies.seekers[k].status == ObjectStatus.Dying)
        {
            Game.IncreaseLife(playerId);
            Game.SeekerCrashed((byte)k);
        }
    }
}
}
}

```

You may have noticed the `DetectCrash` and `DetectCollision` method calls in the code above. These two helpers functions do the real work of determining if a player or a bullet has hit an enemy. To see if a bullet collides with an enemy, we treat the bullet as a point and the enemy as a rough circle. Testing if a point is within a circle is pretty straightforward:

From `GamePlayHost.cs`:

```

public static bool DetectCollision(Vector3 bullet, Vector2 enemy,
    double totalGameSeconds)
{
    Vector2 pos =
        GamePlayHost.FindBulletPosition(bullet, totalGameSeconds);
    if (Vector2.Distance(pos, enemy) < 30)
        return true;

    return false;
}

```

To see if a player collides with an enemy, we treat each object as a rough circle. Then to see if the two circles touch, we add the radius of each circle together, and compare that with the real distance between the center of each circle:

From `GamePlayHost.cs`:

```

public static bool DetectCrash(Avatar player, Vector2 enemy,
    EnemyInfo enemyType)
{
    if (player.status == ObjectStatus.Active)
    {
        float distance = Vector2.Distance(player.position, enemy);
        return (distance < Avatar.CrashRadius + enemyType.CrashRadius);
    }
    return false;
}

```

Game1

Our **Game1** object plays a critical role in this sample. It's responsible for all the gameplay data in the sample, and with few exceptions only the **Game1** object changes the basic state data of the game. It's also in charge of the game's players, hardware configuration, hosting components like the `ScreenManager`, and provides the heartbeat of the game through the `Update` and `Draw` methods. In later versions it will host other managers, like the `NetworkManager`, `InputManager`, and `AudioManager`, as well as controlling network traffic.

Managing Game Data

To signal a change in the state data for the game, `Game1` exposes a series of small methods. We've encountered some of these methods already, like `ShipMove`, `ShipFire`, and `SpawnNextWave`. This state data is kept in three variables, `Game1.enemies`, `Game1.ships`, and `Game1.TotalGameSeconds`. The `Enemies` class keeps track of both enemy waves and the Seekers, and the `Ship` class handles the on-screen players. (A separate class, `Player`, keeps track of the players whether they are currently playing a game or not – specifically, who is signed in to which controller). Here we see an example of how the **Game** object handles data:

From `Game1.cs`:

```
public void SpawnNextWave(float totalGameSeconds)
{
    enemies.SpawnNextWave(totalGameSeconds);
}

public void ShipMove(byte player, Vector2 pos)
{
    ships[player].position = pos;
}

public void ShipFire(byte player, float totalGameSeconds)
{
    ships[player].bullets.Add(new Vector3(ships[player].position,
        totalGameSeconds));
}
```

Managing Players

A robust game has to be able to keep track of players as independent entities, regardless of whether they are playing over the network, or using a local controller (and regardless of which controller they're using). For this reason, we use the `Player` class to keep track of our player information. For now we just track their local controller port, name, and play status (`IsPlaying`). There are two **Player** objects in **Game1 – Main**, and **Guest**. In this sample, the **Main** player controls the menus and becomes **Player One** during any gameplay; the **Guest** player is the second controller and becomes **Player Two**.

We initialize the **Main** player during `Game1.Initialize`. After that, we check during `Update` to see if the **Guest** player has joined the game:

From `Game1.cs`:

```
// Check for Player Two pressing Start, and initialize them.
if (GamePad.GetState(PlayerIndex.Two).IsButtonDown(
    Buttons.Start) && !Guest.IsPlaying)
{
    InitializeGuest(PlayerIndex.Two);
}
```

Menu

Menus are a critical (if boring) component of games. Most games will have more than one, even multiple layers of menus to track. This sample uses two classes to encapsulate the menu. The first class, `MenuComponent`, is a reusable component that will display a list of menu choices and keep track of which item is currently selected. The `MenuComponent` is designed to be hosted by a `GameScreen` that represents the entire menu. In this sample, that `GameScreen` is called `GamePlayMenu`, and is called up by the `GamePlayScreen` when the Main player presses **Start**. The `GamePlayMenu` initializes the `MenuComponent` with a list of menu options along with the current viewport. `GamePlayMenu` also handles the `MenuSelected` and `MenuCanceled` events, which are raised when the user presses the **A** or **B** button:

From `GamePlayScreen.cs`:

```
MenuComponent menu;
...
public override void Initialize()
{
    menu = new MenuComponent(this.ScreenManager.Game,
        this.ScreenManager.Font, this.ScreenManager.SpriteBatch);
}
```

```

menu.AddText("Resume");
menu.AddText(" Help");
menu.AddText("Restart");
menu.AddText(" Quit");
menu.MenuOptionSelected +=
    new MenuEventHandler(menu_MenuOptionSelected);
menu.MenuCanceled += new MenuEventHandler(menu_MenuCanceled);
menu.Initialize();
Viewport view = this.ScreenManager.GraphicsDevice.Viewport;
menu.CenterMenu(view);

TransitionPosition = 0.5f;
base.Initialize();
}

```

To allow the `MenuComponent` to handle input itself we added a `HandleInput` method that uses an `InputState` parameter, just like `GameScreen` objects do. `GamePlayMenu` calls `MenuComponent.HandleInput` during its own `HandleInput`:

From `GamePlayScreen.cs`:

```

public override void HandleInput(InputState input)
{
    menu.HandleInput(input);
    base.HandleInput(input);
}

```

`GamePlayScreen` monitors user input during `Update`, looking for the user to press the **Start** button:

From `GamePlayScreen.cs`:

```

// If the user activates the menu...
if ((GamePad.GetState(game.Main.Controller).Buttons.Start ==
    ButtonState.Pressed) && !bPaused)
{
    this.ScreenManager.AddScreen(menu);
    // Use this to keep from adding more than one menu to the stack
    bPaused = game.BeginPause();
    return;
}

```

That sets in motion a chain of events where we add the `GamePlayMenu` screen to the stack. `GamePlayMenu` displays the `MenuComponent`, which pops up the menu list supplied during `Initialize`.

We also pause the game, and set a Boolean. The Boolean we use is very important. Because we are detecting when the **Start** button is held down, we know for certain that our routine would be called more than once as the **Start** button is pressed down from frame to frame. Whenever you're detecting a button press, you can be quite sure that button will not be down for only one frame. You can wait until the button is released before taking any action, and we do so in other circumstances. However, when it comes to pausing the game, we want the game to appear responsive and not wait for the player to let go of the **Start** button. So in this case we don't wait for the button to be released, and we use the Boolean instead.

Once displayed, the `GamePlayMenu` screen dims the screen, and the `MenuComponent` displays four choices, including an option for Help.

HelpScreen

A **Help** screen is another item that's important for a successful game. Even if your controls are intuitive, and it's clear what the obstacles are and how to overcome them, you can be sure that someone in your audience might be confused or otherwise need a nudge in the right direction. (What if they've never played this type of game before? Or any game?). Even a simple Help screen can go a long way toward satisfying those customers. This sample provides a very simple Help screen that is available from the menu.

Miscellaneous Graphics

The sample project has a file named `Graphics.cs` that contains miscellaneous graphics-related classes.

Background Screen

This game features a background starfield with floating clouds of dust. This theme will be reused for most of our future screens, so we've created the `BackgroundScreen` class to serve as a base class for screens featuring a starfield.

`BackgroundScreen` uses two instances of the `ScrollingBackground` class, each running a full-screen graphic at different speeds.

`BackgroundScreen` also contains a couple of useful utility functions. One, `GetTitleSafeArea`, is designed to calculate the current title safe area. The title safe area is the part of the screen (especially on a standard definition television) that is guaranteed to be visible by the user. Screens should take care to display important graphics and text within this area. The title safe area is typically considered to be the inner 90 percent of the total area.

The other utility function is `DrawBorder`, which draws a colored border around a given rectangle. This is used by many screens in the game to provide either a practical or aesthetic effect. The border is constructed by drawing a single corner graphic in the corners – properly rotated – and using rectangular scaling with `SpriteBatch.Draw` to stretch a single border graphic between the corners.

Explosion

Creating a realistic explosion is hard. Very hard. Nice explosions can add a lot to a game, however (assuming your game has things that explode). Most explosions in games are a multi-frame graphic that shows the explosion begin, expand, and then fade out. In this sense explosions are just another type of animation. There are other ways to render explosions, such as particle systems, especially in 3D. But it's hard to top a real explosion captured digitally and replayed as an animation.

The `Explosion` class takes one frame of an explosion graphic, and use `SpriteBatch` functionality to make it expand, brighten, and then fade away. How large and how bright the explosion becomes is determined by when the explosion starts, when it ends, and the current time. By manipulating one texture we can create the illusion of an animation with a little less overhead than having a string of images that represent an animation.

To create the illusion, we change three parts of the animation: the size, the brightness, and the alpha value. To change the size, we simply create a scale parameter that starts at 60 percent of the original graphic's size, and increases from there:

From `Graphics.cs`:

```
double duration = endSeconds - startSeconds;
double time = totalSeconds - startSeconds;

// The explosion gets bigger as time goes on
float scale = (float) ((time / duration)*.5f) + .6f;
```

Duration represents how long the explosion will last, and time is how long the explosion has already been active. While the explosion is growing, it also gets brighter:

From `Graphics.cs`:

```
// The explosion gets brighter as time goes on
float intensity = (float) (time / duration) * .5f;
Vector4 colorval = new Vector4(0.75f + intensity);
```

The original intensity of the explosion is 75 percent and gets brighter as time goes on. So the explosion starts darker than normal and then grows to full intensity. We accomplish this effect by passing RGB values that are less than 1 as the color parameter of `SpriteBatch.Draw`. This darkens the texture being drawn. In this case, we create a `Vector4` where the X, Y, and Z properties will hold our R, G, and B values.

Part of the color parameter of `SpriteBatch.Draw` is an alpha value. This part of the color determines how transparent the image should be. An alpha value of 0 is fully transparent (i.e. invisible), and 1 is fully opaque. The `Explosion` class uses a modified sine wave to create a smooth increase and decrease in transparency, so the explosion fades in and out. This alpha value is encoded as the W value of the `Vector4` we used to store the RGB values.

From `Graphics.cs`:

```
// The explosion also fades in and out
colorval.W = (float)Math.Sin((time / duration) * MathHelper.Pi)+.2f;
```

Finally, we take the [Vector4](#) and convert it to a Color, where the XYZW values become RGBA values. Then we draw the explosion:

From Graphics.cs:

```
// The intensity and fade determines our tint color
Color color = new Color(colorval);

// Draw the texture with the appropriate scale and tint
batch.Draw(texture, pos, null, color, rotation, origin, scale,
    SpriteEffects.None, 0.5f);
```

Coming up Next

In [Part 2](#) of this series, we'll explore how to move this prototype game into a full-fledged arcade game by adding more screens, audio support, more sophisticated controller and gamer profile handling, and support for gamer presence on Xbox LIVE.

See Also

Related Topics

[TopDownShooter: Polishing the Game](#)

[2D Graphics Overview](#)

[Xbox 360 Programming Considerations](#)

TopDownShooter: Polishing the Game

This article is the second in a series that examines how to construct a complete game using the XNA Framework.

- [The Complete Sample](#)
- [Introduction](#)
- [AudioManager](#)
- [Input Manager](#)
- [Setting Options](#)
- [Networking](#)
- [The Start Screen](#)
- [The SignIn Screen](#)
- [The Menu Screen](#)
- [The Options Screen](#)
- [Changes to GameplayScreen](#)
- [Adding new screens for Game1](#)
- [Integrating the new components into Game1](#)
- [Conclusion](#)

The Complete Sample

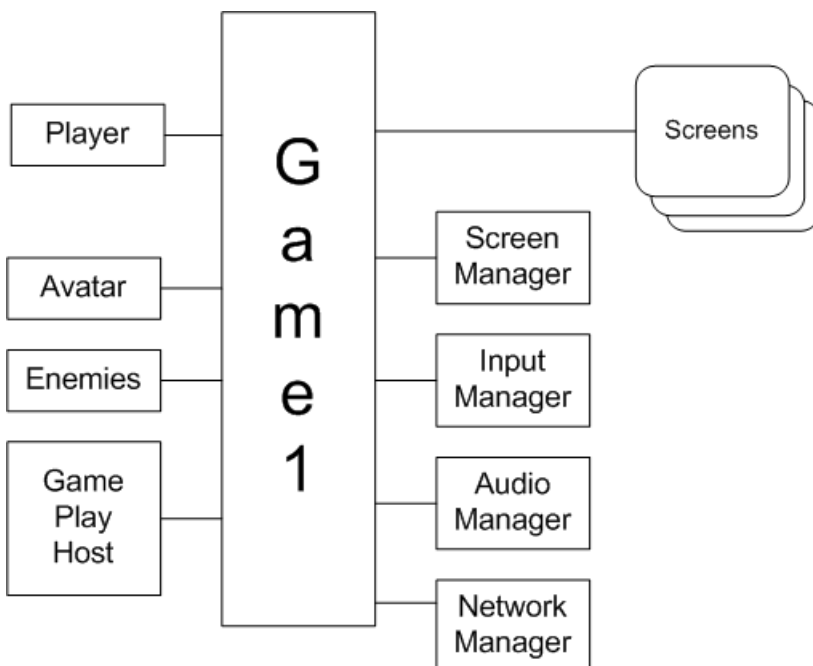
The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download TopDownShooter_2_Sample.zip.](#)

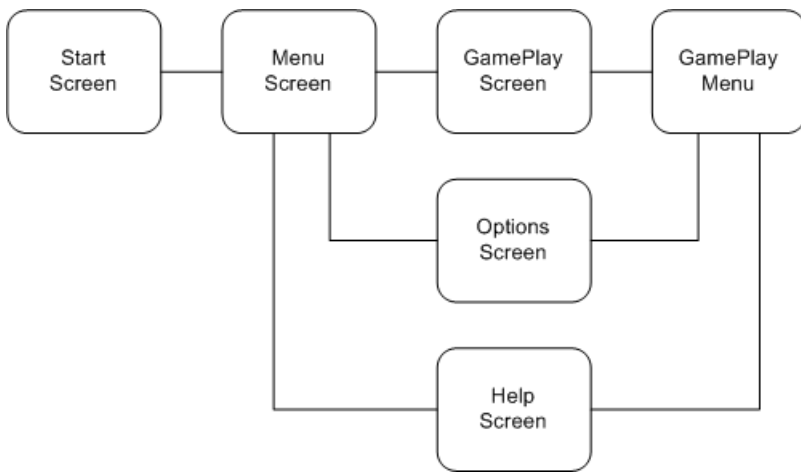
Introduction

In this series, we will touch on not just graphics, but networking, input, storage, audio, and other key topics for making a fully functional game. Along the way, we will show you how to implement features that go beyond gameplay—features such as supporting multiple controller configurations, in-game volume controls, network presence, and other best practices that will make your games appear more polished and professional.

In our second installment, we take the basic game from Part 1, and add audio support, controller support, some Xbox LIVE support, a main menu screen, an options screen, and a start screen. So by the time we are done, the code will look something like this:



We're also adding a main menu screen, an options screen, and a start screen. The options screen and Help screen both exit back to the screen that created them, and the main menu can exit back to the start screen, like so:



AudioManager

To add audio to the sample, we created the `AudioManager`. Like the `ScreenManager`, this is a component that we load and add to the game's `Components` collection. Instead of handling `GameScreens`, however, this component keeps track of our current sounds, and exposes methods we can use to play specific sounds, pause sounds, and set volume levels.

To play audio, the `AudioManager` uses the new `Song` and `SoundEffect` classes. A `SoundEffect` is loaded from the content manager, and encapsulates a wave (WAV) file. For each `SoundEffect`, you have the option of creating a `SoundEffectInstance` for managing the sound after it begins to play. If you choose to use a `SoundEffectInstance`, you must call `SoundEffect.CreateInstance` before you call `SoundEffectInstance.Play`. The `AudioManager` declares a private variable for each `SoundEffect` it uses, and a separate `SoundEffectInstance` for each of the sounds that play continuously. A song can be loaded from the content manager or from the user's existing media library, and the `MediaPlayer` class allows us to play a specific song.

From `AudioManager.cs`:

```

private SoundEffect Shot;
private SoundEffect Explosion;

private Song Music;
private SoundEffect Seeker; // Looped
private SoundEffectInstance SeekerInstance;

```

`AudioManager` is declared as a `DrawableGameComponent`. Even though it never draws, this gives us an overridable `LoadContent` method for loading our sounds:

From `AudioManager.cs`:

```

protected override void LoadContent()
{
    Music = Game.Content.Load<Song>("Menu_Loop");
    MediaPlayer.Volume = MusicVolume;
    MediaPlayer.IsRepeating = true;

    Seeker = Game.Content.Load<SoundEffect>("engine_2");

    // We call CreateInstance to get a permanent handle
    // on the seeker sound, and set it's pitch, volume,
    // and looping accordingly.
    SeekerInstance = Seeker.CreateInstance();
    SeekerInstance.Volume = FXVolume;
    SeekerInstance.Pitch = 0.75f;
    SeekerInstance.IsLooped = true;

    Shot = Game.Content.Load<SoundEffect>("tx0_fire1");
    Explosion = Game.Content.Load<SoundEffect>("explosion1");

    base.LoadContent();
}

```

The `Music` [Song](#) represents the background music, a short sample that plays in a continuous loop. The `Seeker` [SoundEffect](#) represents the sound of the seeker's engine. In our `LoadContent`, we create a [SoundEffectInstance](#) for the `Seeker` sound. Because we want the sound to be continuous when it is played, we set the `IsLooped` property as soon as we get an instance.

Even for a short-lived [SoundEffect](#), it can be useful to hold on to the [SoundEffectInstance](#) returned by `CreateInstance`. We need to have the [SoundEffectInstance](#) if we want to pause and resume the sound. For this game, we want the audio manager to be able to pause all sounds, and resume them. To that end, the `AudioManager` has a queue to hold [SoundEffectInstances](#), and the `AddSound` method that creates a [SoundEffectInstance](#) for a specific [SoundEffect](#), and adds the [SoundEffectInstance](#) to the queue.

From `AudioManager.cs`:

```
private Queue<SoundEffectInstance> Sounds;
private void AddSound(SoundEffect sound, float volume, float pitch)
{
    SoundEffectInstance handle = sound.CreateInstance();
    handle.Volume = volume;
    handle.Pitch = pitch;
    handle.Play();
    Sounds.Enqueue(handle);
}
public void PlayShot()
{
    if (!bPaused)
        AddSound(Shot, 1.0f, 0.5f);
}
```

The methods used to play [SoundEffects](#), like `PlayShot`, all call `AddSound` if the audio is not paused. This adds the [SoundEffectInstance](#) to the queue. During `Update`, the `AudioManager` checks the first sound of the queue. If the sound has expired, it removes it from the queue. This keeps the queue size manageable.

To play music, we use the `Play` method on [MediaPlayer](#).

From `AudioManager.cs`:

```
public void PlayMusic()
{
    MediaPlayer.Play(Music);
}
```

The most useful function for the audio manager is the ability to pause all sounds currently being played. This is why we put each [SoundEffectInstance](#) into a queue. When the audio is paused, we iterate over the queue and pause all the sounds being played. We also pause the `Seeker` sound. To pause everything, including music, we call [MediaPlayer.Pause](#) along with `PauseSounds`.

From `AudioManager.cs`:

```
public void PauseSounds()
{
    bPaused = true;
    SeekerInstance.Pause();
    foreach (SoundEffectInstance item in Sounds)
    {
        if (item.State == SoundState.Playing)
            item.Pause();
    }
}
public void PauseAll()
{
    PauseSounds();
}
```

```
    MediaPlayer.Pause();
}
```

To resume playback, the game calls `ResumeAll`, which iterates over the queue again, resuming any sounds in the `Paused` state. It also calls `Resume` for the `Seeker` instance. If the seeker sound was stopped when `Pause` was called, then the `SeekerInstance` state will be `Stopped` and not `Paused`. Therefore, the seeker sound will not resume playing. To see pause and resume in action, press **Start** during normal gameplay—a popup menu will appear, during which time all sounds are paused. Select **Resume** on the menu to resume the game, and any paused sounds will continue playing.

For a polished game, you probably need to add some kind of sound management tools to pause and resume certain sounds while not pausing others. The `AudioManager` is not a general purpose solution for this problem, but it does illustrate one approach you can take, particularly if you use the `SoundEffect` class. Ultimately, how you manage audio is up to you, the developer.

Input Manager

To manage our controllers for this sample, we created the `InputManager`. The input manager is not responsible for reporting gamepad buttons, triggers, or thumbstick status to the game. That is still handled by the `ScreenManager`, which uses the `InputState` class. The input manager is responsible for just about everything else related to controllers, including vibrating controllers and detecting disconnected controllers.

The first major controller-related task in the game is detecting which controller the main player is using. You may recall this game keeps track of two players with the `Player` class: the `Main` player, who controls the menus and is considered player one, and the `Guest` player, who controls player two.

Most games detect the primary player by presenting a title screen, and asking one of the players to press **Start**. Whoever presses **Start** is designated as player one! Pretty simple. This is the approach we take in our game. The start screen is the first screen displayed by the game. In every frame, the `StartScreen` calls an `InputManager` method, `CheckPlayerOneStart`, to see if anybody has pressed **Start**. Once one of the players presses **Start**, the screen calls `Game1.InitializeMain`, and then exits. `CheckPlayerOneStart` returns the `PlayerIndex` of the player who pressed **Start**, which is passed to `Game1.InitializeMain`.

From `InputManager.cs`:

```
public static bool CheckPlayerOneStart(out PlayerIndex playerOne)
{
    playerOne = PlayerIndex.One;
    if (IsSupported(
        GamePad.GetCapabilities(PlayerIndex.One).GamePadType) &&
        (GamePad.GetState(PlayerIndex.One).Buttons.Start
        == ButtonState.Pressed))
    {
        playerOne = PlayerIndex.One;
        return true;
    }
    if (IsSupported(
        GamePad.GetCapabilities(PlayerIndex.Two).GamePadType) &&
        (GamePad.GetState(PlayerIndex.Two).Buttons.Start
        == ButtonState.Pressed))
    {
        playerOne = PlayerIndex.Two;
        return true;
    }
    if (IsSupported(
        GamePad.GetCapabilities(PlayerIndex.Three).GamePadType) &&
        (GamePad.GetState(PlayerIndex.Three).Buttons.Start
        == ButtonState.Pressed))
    {
        playerOne = PlayerIndex.Three;
        return true;
    }
    if (IsSupported(
        GamePad.GetCapabilities(PlayerIndex.Four).GamePadType) &&
        (GamePad.GetState(PlayerIndex.Four).Buttons.Start
        == ButtonState.Pressed))
    {
        playerOne = PlayerIndex.Four;
    }
}
```

```

        return true;
    }
    return false;
}

```

`CheckPlayerOneStart` relies on a helper function, `IsSupported`, which is defined by the `InputManager`. The `IsSupported` method allows us to define which controllers the game considers valid, and by omission which controllers it should ignore. For example, our game supports a gamepad, an arcade stick, and a flight stick. By extension, we will not support guitars or drums, or `BigButton` controllers. If any of those controllers presses **Start**, we will ignore their inputs.

From `InputManager.cs`:

```

private static bool IsSupported(GamePadType type)
{
    return ((type == GamePadType.GamePad) ||
           (type == GamePadType.ArcadeStick) ||
           (type == GamePadType.FlightStick));
}

```

The second major controller-related task in the game is detecting when the `Main` player's controller is disconnected. That's the first thing the `InputManager` looks for during its `Update` cycle:

From `InputManager.cs`:

```

// Detect disconnected controller
if ((!GamePad.GetState(game.Main.Controller).IsConnected) &&
    (cdScreen == null))
{
    // Display Screen to wait for reconnect
    cdScreen = new ControllerDisconnectScreen(game.Main.Controller);
    game.screenManager.AddScreen(cdScreen);
}

```

The way the input manager handles a disconnected controller is to put a new `GameScreen` – `ControllerDisconnectScreen` – in front of the other screens. This has the effect of interrupting the previous screen. The other screens can detect during `Update` that they are not on top by checking the `coveredByOtherScreen` parameter, and act accordingly. Meanwhile the `ControllerDisconnectScreen` remains on top, displaying an error message, until the user reconnects the controller.

After checking the `Main` controller, the input manager embarks upon its third major task—handling controller vibration. Controller vibration can be a very tricky case because different vibrations are often being exerted at the same time. To accommodate this, `InputManager` defines a `ControllerEvent`, which adds vibration to a particular controller for a specific period of time:

From `InputManager.cs`:

```

public struct ControllerEvent
{
    public int finalFrame;
    public PlayerIndex player;
    public float leftMotor;
    public float rightMotor;
}

```

A method on the `InputManager` called `AddVibration` creates these events on demand and stores them in a **List** object.

From `InputManager.cs`:

```

private int currentFrame = 0;
private List<ControllerEvent> Vibrations;
...
private void AddVibration(PlayerIndex player, int frames,
    float leftMotor, float rightMotor)
{
    ControllerEvent vibe;
    vibe.finalFrame = currentFrame+frames;
    vibe.leftMotor = leftMotor;
    vibe.rightMotor = rightMotor;
    vibe.player = player;
    Vibrations.Add(vibe);
}

```

Finally, during `Update`, the list is combed for obsolete entries, and then each event is tallied for each controller. The input manager then calls the [SetVibration](#) method using the new totals.

From InputManager.cs:

```

if (!bPaused)
{
    currentFrame++;

    Vector2 pOne = Vector2.Zero;
    ...
    // Clear the list of expired events;
    for (int i = 0; i < Vibrations.Count; i++)
    {
        // Add up all the vibrations for each player
        if (Vibrations[i].finalFrame < currentFrame)
            Vibrations.RemoveAt(i);
        else
        {
            switch (Vibrations[i].player)
            {
                case PlayerIndex.One:
                    pOne.X += Vibrations[i].leftMotor;
                    pOne.Y += Vibrations[i].rightMotor;
                    break;
                ...
            }
        }
        // Play the vibrations for each player
        GamePad.SetVibration(PlayerIndex.One, pOne.X, pOne.Y);
        ...
    }
}

```

The `InputManager` also provides a way to pause the vibration, using the `BeginPause/EndPause` pattern common to this program:

From InputManager.cs:

```

private bool bPaused = false;
...
public void BeginPause()
{
    bPaused = true;
    if (GamePad.GetState(game.Main.Controller).IsConnected)
    {
        GamePad.SetVibration(game.Main.Controller, 0, 0);
    }
    if (GamePad.GetState(game.Guest.Controller).IsConnected)

```

```

    {
        GamePad.SetVibration(game.Guest.Controller, 0, 0);
    }
}
...
public void EndPause()
{
    bPaused = false;
}

```

Setting Options

Once the input and audio systems in your game are in place, it's a good idea to think about player options for input and audio. Input options typically include different controller configurations, while audio options normally come down to volume levels for different types of sound. In our sample, we include two controller configurations, and two sound levels. The two controller configurations differ in which thumbstick is used for movement. This is one of the options gamers can specify in their gamertag for cross-game preferences, and so the chance to honor that preference is important. The sound levels apply to sound and music levels.

To model the different options available, we create the `OptionsState` struct:

From OptionsScreen.cs:

```

public enum InputMode
{
    Normal,
    Southpaw,
}
public struct OptionsState
{
    public InputMode PlayerOne;
    public InputMode PlayerTwo;
    public float FXLevel;
    public float MusicLevel;
}

```

The options are stored as a read-only property and a private member on `Game1` called `Options` and `options`, respectively. The first time we run into this structure is when we try to populate it, in a function called `LoadOptions`. `LoadOptions` is called twice—once during `Game1`'s `Initialize` to set sensible defaults, and again when the main player is set for the game. `LoadOptions` deals with two scenarios: if no player is signed in, it looks in the title location for any previously set options; if there is a player signed in, it looks for a valid storage container and loads options from there. If there's nothing to load, it sets some sensible defaults. `LoadOptions` is called twice: once during `Initialize`, resulting in a load from the title container or from sensible defaults, and again when a player presses **Start** on the start screen.

From Game1.cs:

```

public void LoadOptions()
{
    // If we know what the player One storage device is, we call
    // FinishLoadOptions directly. If not, we need to prompt for it,
    // after which the Guide will call FinishLoadOptions for us.
    if ((Main.Device == null) || (!Main.Device.IsConnected))
        Guide.BeginShowStorageDeviceSelector(Main.Controller,
            FinishLoadOptions, null);
    else
        FinishLoadOptions(null);
}
public void FinishLoadOptions(IAsyncResult ar)
{
    // If ar is null, assume we have a device already
    if (ar != null)
        Main.Device = Guide.EndShowStorageDeviceSelector(ar);
}

```

```

// In case the user didn't supply a valid storage device, check
// the title location
string path = Path.Combine(StorageContainer.TitleLocation,
    "options.xml");

// If they did, open a container
StorageContainer container = null;
if (Main.Device != null)
{
    container = Main.Device.OpenContainer("TopdownShooter");
    path = Path.Combine(container.Path, "options.xml");
}

if (File.Exists(path))
{
    XmlSerializer serializer = new XmlSerializer(typeof(OptionsState));
    FileStream file = File.OpenRead(path);
    options = (OptionsState)serializer.Deserialize(file);
    file.Close();
}
else // If there's no options to find, set sensible defaults
{
    options = new OptionsState();
    options.FXLevel = 0.7f;
    options.MusicLevel = 0.6f;
    options.PlayerOne = InputMode.Normal;
    options.PlayerTwo = InputMode.Normal;
}

// Dispose the container if we opened one:
if (container != null)
    container.Dispose();

// Set the audio manager with the options we just loaded
audio.SetOptions(options.FXLevel, options.MusicLevel);
}

```

This is our first foray into the [Microsoft.Xna.Framework.Storage](#) namespace. Because we need to call [Guide.BeginShowStorageDeviceSelector](#) and that's an asynchronous function, the function is broken into pieces: `LoadOptions` and `FinishLoadOptions`. [BeginShowStorageDeviceSelector](#) will call this function for us when it completes, if we don't end up calling it directly. `FinishLoadOptions` builds a path based on whether we have a `StorageDevice` defined for us (meaning the player chose a device instead of cancelling the dialog), and then loads a serialized version of `OptionsState` from a file named `options.xml`.

In some cases, there won't be options to load, so this is where we encode our default settings and return those. Regardless of where the options come from, we immediately apply them.

When calling a [Guide](#) function like [BeginShowStorageDeviceSelector](#), it's important to consider what your game will do if the user cancels the dialog instead of choosing what you expect the user to choose. Presenting the same dialog again will probably result in another cancellation, so it's good to either have a backup plan (in this case, we look in the title container, or load defaults) or be prepared to quit whatever process required information from the user. We also double-check that the last device specified is still there when we try to access it, and prompt to replace it if it's gone. If they're using a memory unit, this has the beneficial side effect of reminding them to re-insert it.

After a player uses the `OptionsScreen` to change the options, it calls `SetOptions`, which acts as a mirror image to `LoadOptions`:

From Game1.cs:

```

public void SetOptions(OptionsState state)
{
    this.options = state;

    // Reassign gamer preferences
    Main.Options = state.PlayerOne;
    Guest.Options = state.PlayerTwo;

    // Change audio levels

```

```

audio.SetOptions(state.FXLevel, state.MusicLevel);

// If we know what the player One storage device is, we call
// FinishSetOptions directly. If not, we need to prompt for it,
// after which it will call FinishSetOptions for us.
if ((Main.Device == null) || (!Main.Device.IsConnected))
    Guide.BeginShowStorageDeviceSelector(Main.Controller,
        FinishSetOptions, null);
else
    FinishSetOptions(null);
}
private void FinishSetOptions(IAsyncResult ar)
{
    // If ar is null, assume we have a device already
    if (ar != null)
        Main.Device = Guide.EndShowStorageDeviceSelector(ar);

    // In case the user prompted did not supply a valid storage device, save
    // options to title storage.
    string path = Path.Combine(StorageContainer.TitleLocation, "options.xml");
    StorageContainer container = null;
    // If they did supply a valid storage device, save options in a container
    if (Main.Device != null)
    {
        container = Main.Device.OpenContainer("TopdownShooter");
        path = Path.Combine(container.Path, "options.xml");
    }

    XmlSerializer serializer = new XmlSerializer(typeof(OptionsState));

    FileStream file = File.Open(path, FileMode.Create);
    serializer.Serialize(file, this.options);
    file.Close();
    if (container != null)
        container.Dispose();
}

```

Networking

Even though this version of our sample is a single-player game, there are a few useful things you can do with networking. For our sample, we use LIVE to set **Presence Information**, which is one of a few pre-defined strings that show up as the player's online status—for example, **At Menu**, **Local CoOp**, and so on. To accommodate setting presence strings, we made a few additions to the `Player` class:

From Player.cs:

```

GamerPresenceMode previousMode;
public void BeginPause()
{
    if (SignedInGamer != null)
    {
        previousMode = this.SignedInGamer.Presence.PresenceMode;
        this.SignedInGamer.Presence.PresenceMode = GamerPresenceMode.Paused;
    }
}
public void EndPause()
{
    if (SignedInGamer != null)
        this.SignedInGamer.Presence.PresenceMode = previousMode;
}
public GamerPresenceMode SetPresence(GamerPresenceMode mode)
{
    if (SignedInGamer != null)
    {
        GamerPresenceMode retval = this.SignedInGamer.Presence.PresenceMode;
        this.SignedInGamer.Presence.PresenceMode = mode;
        return retval;
    }
}

```



```

        return GamerPresenceMode.None;
    }
    public int SetPresenceValue(int value)
    {
        if (SignedInGamer != null)
        {
            int retval = this.SignedInGamer.Presence.PresenceValue;
            this.SignedInGamer.Presence.PresenceValue = value;
            return retval;
        }
        return 0;
    }
}

```

BeginPause and EndPause are both called by Game1 to swap the PresenceMode to "Paused" and back. SetPresence is called by Game1 during Update based on the PresenceMode property of the topmost game screen:

From Game1.cs:

```

// Which type of screen is on top?
GamerPresenceMode mode =
    this.screenManager.GetScreens()[0].PresenceMode;
Main.SetPresence(mode);
Guest.SetPresence(mode);

```

PresenceMode is a new property we added to GameScreen for this project. Because the PresenceMode largely depends on what screen the player is currently using, it made sense to let the screens dictate the mode.

From GameScreen.cs:

```

public GamerPresenceMode PresenceMode
{
    get { return presenceMode; }
    protected set { presenceMode = value; }
}

GamerPresenceMode presenceMode = GamerPresenceMode.None;

```

The GameplayScreen uses [GamerPresenceMode.Score](#), which requires a presence value to indicate the score. That score is set separately by the Avatar class by using its Update method. The Avatar class has a Player property that gives it access to the SetPresenceValue method:

From Avatar.cs:

```

// Set presence info for Player
Player.SetPresenceValue(score);

```

This means that while GameplayScreen is active, users on Xbox LIVE who look at the player's online status will see their current score!

The Start Screen



Three new main screens have been added to this example in order to provide more options and create a normal Xbox experience. The first screen is the start screen. The start screen is the title screen that most games present after the developer and publisher logos. Players are asked to press the **Start** button, and when they do the screen transitions to the main menu. Thus, the attractive title screen serves the purpose of identifying the controller being used by the main player.

The `StartScreen` class is based on the `BackgroundScreen` class, so it will display the starfield common to the rest of the game. On top of the starfield, the `StartScreen` has a fairly simple `Draw` routine that draws the title of the game with a flashing **Press Start** message beneath it. The `StartScreen` checks every `Update` to see if anyone has pressed start (using the `InputManager`'s `CheckPlayerOneStart` method). If one of the controllers does press **Start**, it sets an exiting flag and calls `TrySignIn` on `Game1`:

From `Game1.cs`:

```
public void TrySignIn(ScreenFinished handler)
{
    // Prompt for sign in if nobody is signed in
    if (SignedInGamer.SignedInGamers.Count == 0)
    {
        SignInScreen screen = new SignInScreen(1, true);
        screen.ScreenFinished += handler;
        screenManager.AddScreen(screen);
    }
    else
        handler();
}
```

`TrySignIn` creates a popup screen, `SignInScreen`, that calls [Guide.ShowSignIn](#) and remains up as long as the guide is present.

The SignIn Screen

The sign-in screen is a brief screen that calls [Guide.ShowSignIn](#) to prompt a user to sign in. This screen will remain on top as long as the guide is showing, then invoke a callback function (if one was supplied) and exit. Most of the logic in `SignInScreen` happens in `Update`:

From `NetworkManager.cs`:

```
bool GuideShown = false;
public override void Update(GameTime gameTime, bool otherScreenHasFocus,
    bool coveredByOtherScreen)
{
    // If we haven't activated the guide yet,
    // and it's not up for another purpose, activate it now
    if ((!GuideShown) && (!Guide.IsVisible))
    {
        Guide.ShowSignIn(paneCount, onlineOnly);
        GuideShown = true;
    }
}
```

```

else if ((GuideShown) && (Guide.IsVisible))
{
    // If the guide is up, do nothing
}
// Screen must have been shown and closed
else if (!Guide.IsVisible)
{
    // Activate our callback function and exit
    if (ScreenFinished != null)
        ScreenFinished();
    ExitScreen();
}
base.Update(gameTime, otherScreenHasFocus, coveredByOtherScreen);
}

```

The Menu Screen



After presenting the start screen and prompting for sign-in, the game presents the main menu screen, to which the user will return until he or she quits the game. The `MenuScreen` is based on the `BackgroundScreen` class, and uses the same menu component featured in the first sample. In this case the `MenuScreen` provides two strings for each menu item, and the `MenuComponent` is smart enough to display the second string for the item that is currently selected. As you can see from the code where the `MenuComponent` is initialized, the main menu provides four options: **Local Game**, **Options**, **Help**, and **Quit**:

From `MenuScreen.cs`:

```

menu = new MenuComponent(this.ScreenManager.Game,
    this.ScreenManager.Font);

//Initialize Main Menu
menu.Initialize();

menu.AddText("  Local Game",
    " - Up to two players on this console");
menu.AddText("    Options",
    " - Change volume levels or controller configuration");
menu.AddText("      Help",
    " - How to play");
menu.AddText("    Quit",
    " - Return to Arcade");
menu.uiBounds = menu.GetExtents();
menu.uiBounds.Offset(uiBounds.X, titleBounds.Bottom + 60);
menu.SelectedColor = Color.LightBlue;
menu.MenuOptionSelected += new MenuEventHandler(menu_MenuOptionSelected);
menu.MenuCanceled += new MenuEventHandler(menu_MenuCancelled);

```

When the main player selects a menu item, the `MenuScreen` takes action. This occurs in the event handler for the `MenuComponent`:

From `MenuScreen.cs`:

```
void menu_MenuOptionSelected(int selection)
{
    switch (selection)
    {
        case 0: // Local Game
            ExitScreen();
            ((Game1)this.ScreenManager.Game).BeginSinglePlayer();
            break;
        case 1: // Options
            ((Game1)this.ScreenManager.Game).DisplayOptions();
            break;
        case 2: // Help
            ScreenManager.AddScreen(new HelpScreen());
            break;
        case 3: // Quit
            this.ScreenManager.Game.Exit();
            break;
        default:
            break;
    }
}
```

The `MenuScreen` remains resident in memory as other screens are spawned on top of it. As they call `ExitScreen`, `MenuScreen` becomes the topmost screen and takes over input processing.

The Options Screen



With the addition of sound and controller options, we need to create a way for users to change these options. So we've created a very simple screen called `OptionsScreen`. The options screen is available from the main menu, and from the popup menu provided by `GamePlayScreen` during a game. The screen is exactly the same in each case.

The options screen divides the display in half. In the top half, a drawing of the Xbox controller is shown, with a legend indicating either the Normal or Southpaw button arrangements. Which legend is displayed depends on the controller layout of the main player. As the menu selection changes between the `Main` and `Guest` player's controller configuration, the legend changes to match. If either configuration is changed, the legend changes to match as well. The legends themselves are just static graphics that overlay the controller graphic.

In your own game, it is a very good idea to display a graphic with a picture of the controller (or more than one) accompanied by a legend showing which buttons and controls are used, and for what purpose. If your game uses triggers and shoulder buttons, you may even want to show two graphics, so the second can highlight those controls. Stock images of the controller are available on the [XNA Creator's Club](#) Web site. As mentioned previously, this game also provides two controller configurations, to allow us to respect the `MoveWithRightThumbstick` preference that can be found on the [SignedInGamer's](#)

[GameDefaults](#) property.

Below the controller options are two more options for music and sound volume levels. These options use the `SliderComponent` class, a small [DrawableGameComponent](#) that displays a series of lightened and darkened bars indicating a numeric value. The `SliderArea` property is a rectangle indicating the drawable area for the component. `SliderUnits` sets the total number of bars to display, and `SliderSetting` is the number of those that are lighted. `SetColor` and `UnsetColor` determine the lighted and unlighted color, respectively.

From OptionsScreen.cs:

```
musicSlider = new SliderComponent(this.ScreenManager.Game,
    this.ScreenManager.SpriteBatch);
musicSlider.Initialize();
tempExtent = menu.GetExtent(3);
musicSlider.SliderArea = new Rectangle(menu.uiBounds.Right + 20,
    tempExtent.Top + 4, 120, tempExtent.Height - 10);
musicSlider.SliderUnits = 10;
musicSlider.SliderSetting =
    (int)(state.MusicLevel*musicSlider.SliderUnits);
musicSlider.SetColor = Color.Cyan;
musicSlider.UnsetColor = Color.DodgerBlue;
```

For this screen, the `MenuComponent` does not handle all its own input. Instead, we use the `HandleInput` method on the screen itself to process input based on the current selection. During `HandleInput`, we check the current selection, and call a specific handler method for each menu item to process input as it sees fit:

From OptionsScreen.cs:

```
public override void HandleInput(InputState input)
{
    // If they press Back or B, exit
    if (input.IsNewButtonPress(Buttons.Back) ||
        input.IsNewButtonPress(Buttons.B))
        ExitScreen();

    musicSlider.UnsetColor = menu.UnselectedColor;
    volumeSlider.UnsetColor = menu.UnselectedColor;
    switch (menu.Selection)
    {
        case 0:
            HandlePlayerOne(input, Game.Main.Controller);
            break;
        case 1:
            HandlePlayerTwo(input, Game.Main.Controller);
            break;
        case 2:
            HandleFXAudio(input, Game.Main.Controller);
            volumeSlider.UnsetColor = menu.SelectedColor;
            break;
        case 3:
            HandleMusic(input, Game.Main.Controller);
            musicSlider.UnsetColor = menu.SelectedColor;
            break;
        default:
            break;
    }

    // Let the menu handle input regarding selection change
    // and the A/B/Back buttons:
    menu.HandleInput(input);

    base.HandleInput(input);
}
```

The `MenuComponent`'s `Update` method still handles changes to the selection, and will raise the `MenuOptionSelected` event if the player presses the **A** button. If that happens, `OptionsScreen` only responds if the current selection is **Save and Exit**:

From `OptionsScreen.cs`:

```
void menu_MenuOptionSelected(int selection)
{
    if (menu.Selection == 4) // Save and Exit
    {
        ExitScreen();
        Game.SetOptions(state);
    }
}
```

This is also where we tell `Game1` what the new options are, as we exit.

Changes to `GamePlayScreen`

You may remember that our prototype `GamePlayScreen` had a popup menu allowing people to exit, restart, and so on. Because we now have a main menu, we want users who exit the game to go back to the main menu, not exit the game. We also want to give them access to the `HelpScreen` and the `OptionsScreen` from within the game. That means modifying `GamePlayMenu` with some new menu options in the `Initialize` function:

From `GamePlayScreen.cs`:

```
public override void Initialize()
{
    menu = new MenuComponent(this.ScreenManager.Game, this.ScreenManager.Font,
        this.ScreenManager.SpriteBatch);
    menu.AddText("Resume");
    menu.AddText("Help");
    menu.AddText("Options");
    menu.AddText("Restart");
    menu.AddText("Quit to Main Menu");
    menu.MenuOptionSelected += new MenuEventHandler(menu_MenuOptionSelected);
    menu.MenuCanceled += new MenuEventHandler(menu_MenuCanceled);
    menu.Initialize();
    Viewport view = this.ScreenManager.GraphicsDevice.Viewport;
    menu.CenterMenu(view);

    TransitionPosition = 0.5f;
    base.Initialize();
}
```

Because we have new menu options, the callback function that responds to the menu has to be updated, too:

From `GamePlayScreen.cs`:

```
void menu_MenuOptionSelected(int selection)
{
    switch (selection)
    {
        case 0: // Resume
            break;
        case 1: // Help
            ScreenManager.AddScreen(new HelpScreen());
            break;
        case 2: // Options
            game.DisplayOptions();
            break;
        case 3: // Restart
            game.Restart();
            break;
    }
}
```

```

        case 4: // Quit
            QuitToMenu();
            break;
        default:
            break;
    }
}

```

Adding new screens for Game1

Now that we have the new screens, we have to integrate them into `Game1`. When `Game1` has finished initialization, it loads the `StartScreen` before exiting `Initialize`:

From `Game1.cs`:

```

// Bring up the Start screen
screenManager.AddScreen(new StartScreen(this));

```

`StartScreen` loads `MenuScreen` itself before it exits, but `StartScreen` will also call `Game1.InitializeMain` to setup the `Main` player with the information that we can gather from his or her profile:

From `Game1.cs`:

```

public void InitializeMain(PlayerIndex index)
{
    if (!Main.IsPlaying)
    {
        SignedInGamer gamer = NetworkManager.FindGamer(index);
        if (gamer == null) // No signed in gamer on this controller
            Main.InitLocal(index, "Player One", InputMode.Normal);
        else
        {
            Main.InitFromGamer(gamer);
            LoadOptions();
        }
    }
}

```

If the network manager can find a gamer associated with that controller, we initialize `Main` with the gamer's information, and load options from the player's container.

After the main menu is active, it will load screens or exit with no interaction with `Game1` until it tries to start a game. To start a game, it calls `BeginSinglePlayer`:

From `Game1.cs`:

```

public void BeginSinglePlayer()
{
    Reset();

    // Get our players and ships setup properly
    ships[0].Activate(Main);
    ships[1].Player = Guest;
    bStateReady = true;

    host.StartGame();
    playScreen = new GameplayScreen(this);
    screenManager.AddScreen(playScreen);
}

```

At this point, `Game1` leaps into action. It assigns players to ships, it starts the `GamePlayHost`, and it launches the `GamePlayScreen` itself.

Integrating the new components into `Game1`

Once the game is underway, some of the gameplay events are modified to play sounds and vibrations. For example, when the ship fires, there's a sound and a controller vibration associated with it:

From `Game1.cs`:

```
public void ShipFire(byte player, float totalGameSeconds)
{
    ships[player].bullets.Add(new Vector3(ships[player].position,
        totalGameSeconds));
    audio.PlayShot();

    input.PlayShot(player);
}
```

To get this functionality, of course, we have to add these components to the `Game1` constructor:

From `Game1.cs`:

```
// This gives us access to the guide
Components.Add(new GamerServicesComponent(this));

// This component handles audio playback
audio = new AudioManager(this);
Components.Add(audio);

// This component keeps track of controllers for us
input = new InputManager(this);
Components.Add(input);
```

Conclusion

Now that we've added support for Xbox gamers, multiple controller types, sound and input options, we've taken our prototype game to a more professional level using the XNA Framework. This project follows many of the best practices for Xbox LIVE Indie Games, and will hopefully guide you to making polished and successful Indie Games of your own using the XNA Framework!

See Also

Related Topics

[TopDownShooter: Creating the Prototype](#)

[Audio Overview](#)

[Storage Overview](#)

[Gamer Services Overview](#)

FuelCell

Discusses the FuelCell 3D game tutorial.

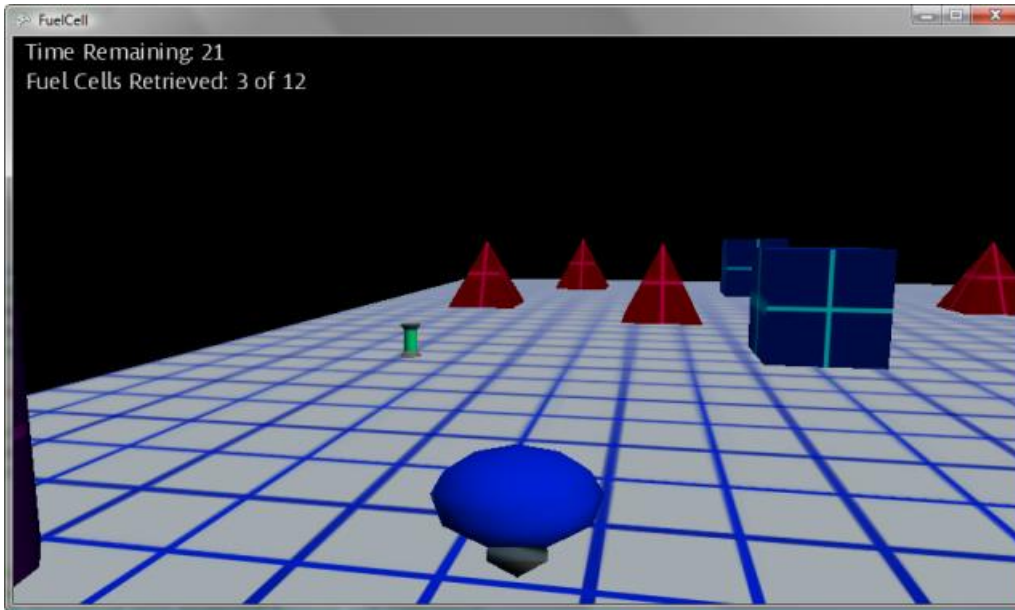


Figure 1. The Complete FuelCell Game

The purpose of this multipart topic is to educate you about various strategies and challenges encountered during the development of a simple 3D game. The FuelCell diary focuses on developing a basic 3D game with a simple goal: collecting objects scattered randomly around a playing field, which is also populated with randomly-placed barriers. The components of FuelCell are deliberately limited to provide a clear picture of the structure and scope of the game.

Game development is hard – especially 3D game development. XNA Game Studio was developed specifically to make game development easier and, hopefully, more enjoyable. Unlike the solid content found in the [Programming Guide](#) and [Getting Started with XNA Game Studio](#) sections, the purpose of this diary is to provide a continuous discussion of the work required for developing a simple 3D game using XNA Game Studio. In addition to articles focused on implementing a specific technique, this diary also contains material on initial game design and how game design drives future development of that game. Also, since game development is not all candy and roses, the diary also focuses on problems encountered during development and how they were overcome.

FuelCell is designed around a central concept: a basic gameplay mechanism that is easily extensible. The initial version, discussed in the following topics, draws a random playfield of barriers and collectable objects. The player, controlling a vehicle that can be driven around the playing field, searches for a collection of fuel cells. Due to the modularity of the game design, future versions could easily add additional effects and game features.

The primary components of FuelCell (camera, avatar control, and collision detection) use existing topics found in the [Programming Guide](#). This diary ties together these disparate concepts into a complete (albeit, simple) 3D game.

Each step in the series contains a list of important topics that are covered in the body of the topic and a link to the completed sample code of that step.

In This Section

[FuelCell: Game Design](#)

Discusses the importance of designing a game using a feature-based approach.

[FuelCell: Setting the Scene](#)

Discusses the implementation of a playing field for the game and a simple, fixed camera.

[FuelCell: Casting Call](#)

Discusses the implementation of the remaining game elements, such as barriers, fuel cells, and an avatar model.

[FuelCell: What's My Motivation](#)

Discusses the implementation of user control for the avatar (known as the fuel carrier).

[FuelCell: What's My Line?](#)

Demonstrates a simple, random technique for placing barriers and fuel cells on the playing field.

[FuelCell: "Ships" Passing in the Night](#)

Discusses collision detection in a 3D game and demonstrates basic collision checking between the fuel carrier vehicle and the other game objects on the playing field.

FuelCell: Finishing Touches

Discusses the final steps in the development of the FuelCell game.

FuelCell: Game Design

Discusses the importance of designing a game using a feature-based approach.

The main points of this topic are:

- Importance of Game Design

The Purpose Behind Game Design

Before every good game, there was game design. Notice that part about "good games"? A common pitfall for game developers is that they might have great ideas in mind, but the translation to the screen turns out to be impossible. It's hard not to lose heart when the developer must focus on developing basic game and code structure before displaying the defense of a planet against hordes of 3D alien enemies. Game design can seem boring and needless when compared to getting the developer's vision onto the monitor.

Unfortunately, a poor game design can lead to the situation where a person sits down with the latest demo, plays it for ten minutes, and gets lost or frustrated.

Like computer games, books are another example of a product that requires a careful design. Every good book has a "hook" line: an opening sentence that immediately piques the reader's curiosity or interest. If the reader finishes the first page (or even the first paragraph) and does not have at least one story-related question rattling around in the brain, that reader probably will not finish the book.

Game design is a technique that creates the "hook" for the player. Your purpose is to capture the gamer's attention in the first couple of minutes. Otherwise, the player may walk away. A good set of game features enrolls your audience into the game's story. It convinces a player to finish the game. However, a good opening is not enough. You must have equally good content that continues to engage or challenge the gamer.

With that in mind, in addition to a good opening, games typically benefit from the following features:

- A clear and achievable goal. But what about the cries of outrage from the 80's arcade crowd: "And what was the achievable goal of <insert favorite 80's arcade game here>?! There was no goal – you died no matter how good you were!" Okay, they may have a point there, but times have changed. Today's games no longer require a constant flow of quarters to survive; they require a consistent paycheck (or a consenting adult with a consistent paycheck).
- Replayability. This can range from a randomly-generated dungeon level or treasure pile to an ending that depends on the actions of the player. It can be argued that replayability is not a requirement of a good game, but it helps to lengthen the span of interest in a game, and it provides more entertainment for the player without the need for extra content or expansion packs.
- One or more obstacles to overcome. No one wants a game where the player just wanders over to the end level boss, it falls dead at your feet, and the end credits roll. A little challenge (or a lot) is necessary to keep the player engaged. This can be, and usually is, the main content of a game. Common challenges include collecting special items, defeating enemies to reach a specific place in the game, or solving a complex puzzle using simple in-game items collected (or taken by force) earlier.

If a game provides at least these features, it is probably a game worth playing at least once. This is the part where game design comes into play. When you design your game, keep these features in mind. Providing content for these features is difficult, but worth the effort when it all comes together.

FuelCell Features

To illustrate this point, the game design for FuelCell focused on fulfilling these three criteria:

Game Feature	FuelCell Feature

Client	Collect 12 fuel cells (green canisters of goo) before time runs out. Fuel cells are collected using a player-controlled model (the fuel carrier). The game is won when the player collects all fuel cells and time remains. The game is lost if less than 12 fuel cells are found and time runs out.
Goal	
Requirements	The playing field is a basic, level grid with fuel cells scattered at random. In addition, a large amount of barriers are also scattered about the playing field. These barriers are opaque and (eventually) cannot be driven through. The replayability comes from the random placement of the fuel cells and barriers for each new game.
Challenges	The fuel carrier always starts in the middle of the field, but the fuel cells and barriers are scattered randomly. There are enough barriers to prevent the player from initially seeing all available fuel cells. The barriers have two purposes: slowing the player down and obscuring some of the fuel cells at all times. In some cases, a fair bit of hunting is required by the player to retrieve all fuel cells. In later versions of the game, existing factors could easily be tweaked to increase the difficulty and introduce new sub-goals (such as a power-up allowing a clearer view or a faster vehicle).

With the FuelCell feature set explained, let's move on to the actual coding of the game. The next [step](#) focuses on a critical part of any 3D game – the camera.

See Also

Conceptual

[FuelCell](#)

FuelCell: Setting the Scene

Discusses the implementation of a playing field for the game and a simple, fixed camera.

The main points of this topic are:

- Implementation and Positioning of a 3D Camera (illustrated by [How To: Make a First-Person Camera](#))
- Drawing 3D Models (illustrated by [How To: Render a Model](#))
- Adding 3D Content to the Application

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download FuelCell_2_Sample.zip.](#)

Note

You must download the above sample code in order to access the 3D models used in this tutorial step.

Overview

One of the largest hurdles a game developer faces when moving from 2D to 3D is that third 'D': depth. In the 2D world, game objects (called sprites) have two dimensions and are positioned using literal screen coordinates. There is a concept of depth, but this is used only to determine if a sprite is partially or fully obscured by another.

In a 3D game, what you see on your screen is a projection of a 3D environment onto a 2D surface (that is, your screen). This translation of 3D space into 2D space is accomplished using transformation matrices. Specifically, we refer to these three matrices as world, view, and projection matrices. Transformation is just a fancy word for changing the value of a coordinate by multiplication. Using these matrices, the XNA Framework transforms the coordinates of a 3D model to a set of new coordinate values (through rotation, scaling, or translation) used by the projection matrix. In a separate but related step, a view matrix simulates a viewpoint (often called the camera) in the same 3D space as the model; it looks in a certain direction. With these two matrices, a third matrix is brought into the "picture" to perform a final transformation into 2D screen coordinates. This creates a realistic 2D picture of the 3D scene on your computer screen.

Earlier, we mentioned a camera. Even though this isn't a real camera, it fulfills the same role in the 3D game. This camera observes the 3D world and renders whatever it sees into a 2D representation. This representation appears on the computer screen. In a game, the camera class usually is implemented as a stand-alone class. It is one of two varieties: a first-person camera (used in this game and first-person shooters) and a third person camera (often used in RPGs or platform games). First-person cameras are great for games that focus on a single player or are trying to immerse the player in the game world. Third-person cameras are better suited to viewing a large playing field or controlling numerous entities in the game. For this step, you will implement a first-person camera using code from [How To: Make a First-Person Camera](#).

We use a first-person camera because the player controls a small vehicle that can move around and collect fuel cells. The difficulty of the game is finding these items before time runs out. It's difficult because the playing field has opaque barriers randomly scattered across it. Since we use a first-person camera, the player must drive around to view previously-hidden areas.

In addition to the camera code, you will also use code from [How To: Render a Model](#) to display the 3D playing field model, which is a simple two-tone grid floating in space.

Objects in the Game

3D game development is all about position and the relation to other objects in the local coordinate system (that is, the game world). In addition to position, a 3D object usually has an associated model. Because this is a 3D game, the model has three dimensions. This means it can be viewed from all angles and has volume. In addition to these two properties, the 3D object should have a bounding sphere. The bounding sphere is a theoretical sphere that encapsulates the model volume. It is used for detecting collisions in the game world with other 3D objects. You can ignore this for now, but it becomes critical later in the development process.

A class is the obvious solution for storing and tracking all these variables. However, before we can add this class, you need to first create a new project for the FuelCell game.

- Open XNA Game Studio and create a Windows Game 3.1 project called FuelCell.

- In this project, create a new class called `GameObject`.

The `GameObject` class will contain all those properties mentioned earlier and a constructor that sets the various properties to known values. The file containing this new class only has a few references by default (located at the top). To grant easy access to the XNA Framework assemblies, you'll need to add some XNA-specific ones. At the top of the file, add the following references:

C#

```
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
```

These new references make it possible to use the short form of reference for XNA Framework-specific classes. For instance, instead of declaring a variable of type `Vector2` by specifying `Microsoft.Xna.Framework.Vector2`, we can use `Vector2` instead. That will save a lot of typing during the development of `FuelCell`!

You're ready to modify the default class declaration to better fit your needs. Replace the existing `GameObject` class declaration with the following:

C#

```
class GameObject
{
    public Model Model { get; set; }
    public Vector3 Position { get; set; }
    public bool IsActive { get; set; }
    public BoundingBox BoundingBox { get; set; }

    public GameObject()
    {
        Model = null;
        Position = Vector3.Zero;
        IsActive = false;
        BoundingBox = new BoundingBox();
    }
}
```

This new version now tracks the position, model, and bounding sphere of an object in the game using auto-implemented properties. The constructor is simple, and it initializes each property to a reasonable value – either **null** or `Vector3.Zero`.

The Camera

The `GameObject.cs` file will also contain the camera class declaration, which is taken from the [How To: Make a First-Person Camera](#) topic. As mentioned earlier, the main purpose behind this developer diary is to demonstrate how you (or any developer) can use various How To articles as stepping stones when developing an XNA Framework game. For this first usage, this concept is clearly illustrated by not changing any of the variable names or classes, whenever possible. This may cause a bit of confusion or head-scratching when you come across variable names like `_avatarHeadOffset` and `avatarYaw`, but it serves to tie the source How To more closely to the actual game code. This creates the ability to easily determine where the source code of a How To ends up in a typical game project by searching for the variable name used in the How To.

For example, in this step, some of the property names match the names used in the original sample code: `_avatarHeadOffset` is the camera's distance above the playing field and `_targetOffset` is the offset from the target. In this case, it is a fixed distance in front of the fuel carrier vehicle. These values are used when calculating the camera position from the current position of the fuel carrier vehicle (for example, `position`) in the world coordinate system.

The camera class is similar in structure to the `GraphicObject` class. It has a set of properties and a method. In this case, it is `Update`. For this game, the camera acts like a rigid chase camera. It follows behind, and slightly above, the actual vehicle and points in the same direction as the vehicle at all times.

All right, enough talk – let's start developing!

Add the following code after the existing `GameObject` class declaration:

C#

```
class Camera
```

```

{
    public Vector3 AvatarHeadOffset { get; set; }
    public Vector3 TargetOffset { get; set; }
    public Matrix ViewMatrix { get; set; }
    public Matrix ProjectionMatrix { get; set; }

    public Camera()
    {
        AvatarHeadOffset = new Vector3(0, 7, -15);
        TargetOffset = new Vector3(0, 5, 0);
        ViewMatrix = Matrix.Identity;
        ProjectionMatrix = Matrix.Identity;
    }

    public void Update(float avatarYaw, Vector3 position, float aspectRatio)
    {
        Matrix rotationMatrix = Matrix.CreateRotationY(avatarYaw);

        Vector3 transformedheadOffset =
            Vector3.Transform(AvatarHeadOffset, rotationMatrix);
        Vector3 transformedReference =
            Vector3.Transform(TargetOffset, rotationMatrix);

        Vector3 cameraPosition = position + transformedheadOffset;
        Vector3 cameraTarget = position + transformedReference;

        //Calculate the camera's view and projection
        //matrices based on current values.
        ViewMatrix =
            Matrix.CreateLookAt(cameraPosition, cameraTarget, Vector3.Up);
        ProjectionMatrix =
            Matrix.CreatePerspectiveFieldOfView(
                MathHelper.ToRadians(GameConstants.ViewAngle), aspectRatio,
                GameConstants.NearClip, GameConstants.FarClip);
    }
}

```

This is the camera class declaration. The major difference between this declaration and its appearance in the original How To is that the camera's functionality has been internalized into a class. This means that previously global variables that tracked camera position, the transformation matrices, and other properties are now stored within the class. These properties can be divided into two parts: offset variables and transform matrices. The offset variables (`AvatarHeadOffset` and `TargetOffset`) force the camera to a specific position behind and above the vehicle's current position. Hence, the name chase camera.

Researching Transformation Matrices

The transformation matrices are used to rotate, move, or scale objects in a world coordinate system and then (along with the view matrix) to a perspective 2D coordinate system: your screen. The theory and application of this concept involves a truckload of math. However, you can read more about these concepts in other areas of the XNA Game Studio documentation:

- [Math Overview](#)
- Step 4 (of [Tutorial 1: Displaying a 3D Model on the Screen](#)) discusses the usage of all three matrices.
- [Viewports and Frustums](#)

The `Update` method is where the main math for updating the camera takes place. This function takes the current rotation of the vehicle and creates a transformation matrix, which in turn is used to transform the camera's offset values. These values are then added to the current vehicle position, creating a point, in the world coordinate system, where the camera "sits." The final step generates the view and perspective matrices, used when rendering the 3D game world view onto your 2D monitor screen.

Game Constants

Did you notice that some of the method arguments were from a `GameConstants` class? Let's create this class and then I'll explain its purpose.

- Add a new class to the project, called `GameConstants`.
- Since you will also be using XNA Framework references in this file, add the following references to the beginning of the

file:

C#

```
using Microsoft.Xna.Framework;  
using Microsoft.Xna.Framework.Content;  
using Microsoft.Xna.Framework.Graphics;  
using Microsoft.Xna.Framework.Input;
```

- Now add the following to the `GameConstants` class declaration:

C#

```
//camera constants  
public const float NearClip = 1.0f;  
public const float FarClip = 1000.0f;  
public const float ViewAngle = 45.0f;
```

You'll use this class to gather common game variables into a single location. You can then easily and quickly alter the value of any game constant and have the new value affect the entire game, or at least those areas where the game constant was used. At this point, you have three candidates for game constants: the near and far clipping planes of the camera and the angle of view used by the camera. The camera's clipping planes determine the distance (in world coordinates) when objects approaching the screen or receding from it are no longer drawn.

It's a good idea to give them informative names so another person, looking at the code, easily understands their purpose.

Okay, that wraps up the camera class and constants implementation. Let's move on to the visually appealing stuff: drawing stuff on the screen!

Getting a Grip

Up until now, the new code has focused on setting up a viewpoint in the game world and added some additional infrastructure that is used by the game and various components. Game assets, in the form of models, are a large part of any 3D game. Even though this is a simple game, `FuelCell` includes many different types of game assets: models that represent game objects, textures that clothe the models, and a font to display game information such as the current score and goal status. For this step, let's add a very basic model and get it on the screen so we can begin to understand how our game world will look.

Every project template created by XNA Game Studio has a sub-project called `Content`. This project must contain all your game assets. Although it isn't required, it's a good idea to organize this content project such that similar assets are in the same folder. A common organization uses several folders: `Models`, `Textures`, `Fonts`, and `Audio`. These folders cover the main parts of a game. Let's add a `Models` folder, and a model, to our game.

⚠Caution

This diary assumes you are using the game assets located in the `FuelCell` sample file downloaded earlier. These assets have been sized in relation to each other so that none are too small or too large. You can use other models, but their scale (the size in the world coordinate system) might be radically different from the `FuelCell` models. This can cause a model to be rendered as a massive or miniscule object in the game world. In some cases, the camera (due to its position) might not be able to see the model at all. Therefore, it is recommended that you use the included `FuelCell` models when following these steps. After gaining some experience working with the camera class and rendering a 3D scene, you can experiment by adding your own models.

- Select the **Content** folder icon and select **New Folder** from the context menu.
- Name this new folder `Models`.
- Select the **Models** folder icon and from the context menu, select **Add** and then **Existing Item...**
- Navigate to the folder containing the downloaded game assets and add the `ground.x` model.

You now have a working camera object, and a ground model, in your project. In the next step, you will add code declaring and initializing both these objects and use them to render a nice terrain in the game world. For the remainder of this step, you will be working exclusively in the `Game1.cs` file, which is the main file of an XNA Framework game.

- Open the `Game1.cs` file using Solution Explorer.
- Add the following code, after the existing declaration of the `graphics` member of `Game1`:

C#

```
GameObject ground;  
Camera gameCamera;
```

- In the existing `Initialize` method, initialize both game objects (using their default constructors) by adding the following code:

C#

```
ground = new GameObject();  
gameCamera = new Camera();
```

- Next, add the following code to the existing `LoadContent` method:

C#

```
ground.Model = Content.Load<Model>("Models/ground");
```

You've added code declaring and initializing your camera class and the terrain model. To see all this work on the screen, you must update the existing `Draw` method to render the terrain. This is also a good time to add code that updates, during each frame, the camera's position and orientation. Currently, this update code does nothing because the fuel carrier (the user-controlled avatar vehicle) isn't in the game yet. However, when the vehicle is added in a later step, the camera automatically updates, chasing the vehicle around as the player tries to find hidden fuel cells.

Opening Your "Eye"

Updating the camera occurs in the aptly-named `Update` method. At this time, the information passed to the `Camera.Update` method is faked because there is no vehicle to focus on. Specifically, the position and rotation for the camera are zeroed out. This means the camera is centered slightly above the terrain model and aligned with the z-axis. This is the axis that represents the depth of the game world. Once you add the vehicle, the `Camera.Update` method will be passed the position and rotation of the vehicle, instead of zeros.

This modification is very simple because you already implemented the `Camera.Update` method. Now, you just need to call it at the proper time and pass some valid values.

- Add the following code to the `Update` method of the `Game1.cs` file:

C#

```
float rotation = 0.0f;  
Vector3 position = Vector3.Zero;  
gameCamera.Update(rotation, position,  
    GraphicsDevice.Viewport.AspectRatio);
```

The final step modifies the existing `Draw` method.

- Modify the body of the `Draw` method of the `Game1.cs` file to match the following:

C#

```
graphics.GraphicsDevice.Clear(Color.Black);  
  
DrawTerrain(ground.Model);
```

This code calls the non-existent `DrawTerrain` method. The method uses the approach detailed in [How To: Render a Model](#) to render the terrain. Let's add that method now.

- Add the following method after the `Draw` method:

C#

```
private void DrawTerrain(Model model)  
{
```

```
foreach (ModelMesh mesh in model.Meshes)
{
    foreach (BasicEffect effect in mesh.Effects)
    {
        effect.EnableDefaultLighting();
        effect.PreferPerPixelLighting = true;
        effect.World = Matrix.Identity;

        // Use the matrices provided by the game camera
        effect.View = gameCamera.ViewMatrix;
        effect.Projection = gameCamera.ProjectionMatrix;
    }
    mesh.Draw();
}
}
```

The `DrawTerrain` method uses a rendering technique commonly used by XNA Framework games – iterative draw calls on child meshes of the parent model. In this rather simple case, the ground model only has one mesh. But for more complex models, this approach is required to properly render the model on the screen. The calls to [EnableDefaultLighting](#) and [PreferPerPixelLighting](#) highlight the power of the XNA Framework because you'll get standard 3-source lighting and smoother model lighting for free, creating some great results with little work!

Go ahead and compile and build your project. You should be hovering over a gray and light-blue terrain under a black sky. It doesn't look like much now, but the next [part](#) adds the rest of the 3D models and displays them on the screen.

See Also

Conceptual

[FuelCell](#)

Tasks

[How To: Make a First-Person Camera](#)

[How To: Render a Model](#)

FuelCell: Casting Call

Discusses the implementation of the remaining game elements, such as barriers, fuel cells, and an avatar model.

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download FuelCell_3_Sample.zip.](#)

Note

You must download the above sample code in order to access the 3D models used in this tutorial step.

Supporting Cast

It's time to add the remaining models: the fuel cells and the fuel carrier. They represent the various barriers encountered in the game.

The fuel cell model (fuelcell.x) is a simple canister-like object with a single texture (fuelcell.png). Typically, you only need to add the model file and not the texture. The texture file is automatically used when the Content Pipeline processes the model file. The barrier models are similar to the fuel cell model. They each have a specific model and a single texture. Since the game has three barrier types, we will be adding three different models (cube10uR/cylinder10uR/pyramid10uR.x) and a set of textures (BarrierBlue/BarrierPurple/BarrierRed.png). Unlike the fuel cell model, the barrier textures are simple and can be used with any barrier model.

Note

The rather unique model names are the result of keeping the scale relatively uniform among all models. The naming convention begins with the model name and then the radius, measured in the units of the 3D modeling application used. Therefore, pyramid10uR is the name of the pyramid model whose radius is 10 units in length.

- Right-click the **Models** directory icon of the Content project.
- Click **Add** and then **New Item....**
- Navigate to the Models sub-directory of the downloaded source, and add the following files:
 - fuelcell.x and fuelcell.png
 - fuelcarrier.x and carriertextures.png
 - cube10uR.x, cylinder10uR.x, and pyramid10uR.x
 - BarrierBlue.png, BarrierPurple.png, and BarrierRed.png
- From Solution Explorer, select all .png files in the Models sub-directory.
- Right-click and select **Exclude From Project**.

This prevents the textures from being processed twice.

Implementing the FuelCell Class

Model implementation is similar to what you did in the last step. You'll add some member variables to store the models, load them with the `LoadContent` method, initialize them (placement, etc.), and then render them on the playing field.

First, the fuel cell class. After the `GameObject` class, add the following `FuelCell` class.

C#

```
class FuelCell : GameObject
{
    public bool Retrieved { get; set; }

    public FuelCell()
        : base()
    {
        Retrieved = false;
    }
}
```

```

public void LoadContent(ContentManager content, string modelName)
{
    Model = content.Load<Model>(modelName);
    Position = Vector3.Down;
}
}

```

Compared to the `GameObject` base class, it contains a new member (`Retrieved`). This is a flag used by the game to determine if the fuel cell has been retrieved. The constructor uses the base constructor, setting `Retrieved` to **false**. You'll use this flag later to optimize your drawing code. If it has been found, it no longer needs to be drawn. The `LoadContent` loads the specified model and then sets the Y-component of the `Position` member to -1. That value is used as an indicator that the fuel cell has not been initialized.

This next bit of code declares a `Draw` method that takes a view and projection matrix and draws the fuel cell. There are two important aspects of this new code: the first is related to optimization and the second is related to future proofing. In terms of optimization, notice the check on the value of the `Retrieved` member. This is the optimization we discussed previously. If the fuel cell has been retrieved by the player, there is no need to draw the fuel cell. This may seem like a minor thing, but changes like this add up over the course of your game's development.

The second is a bit more complicated, but is required for correctly rendering most 3D models: bone transforms. Nearly every 3D model is comprised of a collection of bones. Basically, a model bone is a matrix representing the position of a mesh relative to other bones in the 3D model. An excellent description of model bones can be found [here](#).

For our purposes, it's sufficient to know that a 3D model has a set of these matrices. To position them properly (and, in more complex cases, to animate them), you'll need to incorporate these transform matrices into the code that draws the fuel cell. You can accomplish this easily by copying the bone transforms of the model being drawn into a temporary set of matrices (using [CopyAbsoluteBoneTransformsTo](#)). Then you apply the proper transform matrix to the world matrix before you draw the current sub-mesh of the model. In the code you will add, this is done in the innermost nested **foreach** loop by multiplying the world matrix by the proper transform matrix.

You're probably wondering why this code is called future proofing. Ironically, the code related to applying bone transforms to the 3D model being rendered is unnecessary for `FuelCell`! One of the design rules used for `FuelCell` was to keep everything as simple as possible. This resulted in the rudimentary (and unmoving) models in the game. Each model has a set of bones, but these bones are always rendered statically and cannot be animated. However, this code is necessary to properly render any minimally complex model or one that has movable parts (like a tank rolling across a terrain map). This code makes it easier to be reused in future games without having to remember all this bone transform business. Now, when you copy the code into another 3D game, it just works. Even if the model is very complex!

Now that you understand the code better, add the following code to the `FuelCell` class.

C#

```

public void Draw(Matrix view, Matrix projection)
{
    Matrix[] transforms = new Matrix[Model.Bones.Count];
    Model.CopyAbsoluteBoneTransformsTo(transforms);
    Matrix translateMatrix = Matrix.CreateTranslation(Position);
    Matrix worldMatrix = translateMatrix;

    if (!Retrieved)
    {
        foreach (ModelMesh mesh in Model.Meshes)
        {
            foreach (BasicEffect effect in mesh.Effects)
            {
                effect.World =
                    worldMatrix * transforms[mesh.ParentBone.Index];
                effect.View = view;
                effect.Projection = projection;

                effect.EnableDefaultLighting();
                effect.PreferPerPixelLighting = true;
            }
            mesh.Draw();
        }
    }
}

```

```
}
```

- You'll need to automatically scale the fuel cell model by selecting the newly-added model from Solution Explorer, and, on the property page of the model asset, setting the **Scale** property to .03. The **Scale** property is found by expanding the **Processor** field.

One of the cool features of XNA Game Studio (specifically, the content pipeline) are [processor parameters](#). You can change common values for a processor by changing the related property of a selected game asset using the **Properties** window. If you didn't use this feature, you would need to use a scaling matrix to shrink the fuel cell model before rendering it on the screen.

Did you notice the [Matrix](#) declarations at the beginning of the [Draw](#) function? You need to transform the world coordinates of our object (in this case, the fuel cell) based on the fuel cell's position in the game. If the translation matrix wasn't used before drawing the fuel cell, it would always be in the center of the playing field.

That completes the implementation of the fuel class. Next stop, the barrier class.

Implementing the Barrier Class

The `Barrier` class implements the geometrical barriers that are randomly scattered across the playing field. They are an important part of the game because they provide a new experience for every game (since they are placed randomly) and they provide a challenge to the player who is trying to find fuel cells (also randomly placed) before time runs out. In a later step, when collision detection is added, these barriers become impassable and must be driven around.

- Automatically scale the barrier models by selecting each barrier model from Solution Explorer and setting the **Scale** property to .3, located on the property page of the model asset. The **Scale** property is found by expanding the **Processor** field.

In `GameObject.cs`, add the following class declaration after the `FuelCell` class declaration:

C#

```
class Barrier : GameObject
{
    public string BarrierType { get; set; }

    public Barrier()
        : base()
    {
        BarrierType = null;
    }

    public void LoadContent(ContentManager content, string modelName)
    {
        Model = content.Load<Model>(modelName);
        BarrierType = modelName;
        Position = Vector3.Down;
    }
}
```

The `Barrier` class has a new member (`BarrierType`) that stores, oddly enough, the barrier type. In the `FuelCell` game, there are three possible barrier types: cubes, cylinders, and pyramids. Similar to the `FuelCell` class, the `LoadContent` method is overridden that loads the specified model, stores the barrier type, and sets the Y-component of the `Position` member to -1 (indicating that the barrier is not placed).

Since barrier objects behave in a similar fashion to fuel cells (that is, they stay in one place and do nothing), we'll use the same drawing code with one change. Add the following `Draw` method to the `Barrier` class:

C#

```
public void Draw(Matrix view, Matrix projection)
{
    Matrix[] transforms = new Matrix[Model.Bones.Count];
    Model.CopyAbsoluteBoneTransformsTo(transforms);
    Matrix translateMatrix = Matrix.CreateTranslation(Position);
    Matrix worldMatrix = translateMatrix;

    foreach (ModelMesh mesh in Model.Meshes)
```

```

    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.World =
                worldMatrix * transforms[mesh.ParentBone.Index];
            effect.View = view;
            effect.Projection = projection;

            effect.EnableDefaultLighting();
            effect.PreferPerPixelLighting = true;
        }
        mesh.Draw();
    }
}

```

Unlike fuel cells, barriers can't be retrieved and are always visible. Therefore, there is no need to determine if a barrier should be drawn; it is always drawn. This is reflected in the code.

Implementing the Fuel Carrier

In game development terms, the fuel carrier is the avatar of the player. It is the object that represents the player in the game world and is controlled by the player. The `FuelCarrier` class starts out very simple but, in later steps, you'll add more features like user control and collision detection. For now, it has a few basic methods that load the model and render it on the playing field.

The process for implementing the fuel carrier class is similar to the `FuelCell` and `Barrier` class implementations.

- Automatically scale the fuel carrier model by selecting it from Solution Explorer and setting the **Scale** property to .1, located on the property page of the model asset. The **Scale** property is found by expanding the **Processor** field.
- After the `Barrier` class declaration, add the `FuelCarrier` class declaration:

C#

```

class FuelCarrier : GameObject
{
    public float ForwardDirection { get; set; }
    public int MaxRange { get; set; }

    public FuelCarrier()
        : base()
    {
        ForwardDirection = 0.0f;
        MaxRange = GameConstants.MaxRange;
    }

    public void LoadContent(ContentManager content, string modelName)
    {
        Model = content.Load<Model>(modelName);
    }
}

```

- Implement the `Draw` method by adding the following code to the `FuelCarrier` class declaration:

C#

```

public void Draw(Matrix view, Matrix projection)
{
    Matrix[] transforms = new Matrix[Model.Bones.Count];
    Model.CopyAbsoluteBoneTransformsTo(transforms);
    Matrix worldMatrix = Matrix.Identity;
    Matrix rotationYMatrix = Matrix.CreateRotationY(ForwardDirection);
    Matrix translateMatrix = Matrix.CreateTranslation(Position);
}

```

```

worldMatrix = rotationYMatrix * translateMatrix;

foreach (ModelMesh mesh in Model.Meshes)
{
    foreach (BasicEffect effect in mesh.Effects)
    {
        effect.World =
            worldMatrix * transforms[mesh.ParentBone.Index]; ;
        effect.View = view;
        effect.Projection = projection;

        effect.EnableDefaultLighting();
        effect.PreferPerPixelLighting = true;
    }
    mesh.Draw();
}
}

```

This `Draw` method differs from the fuel cell and barrier `Draw` methods in one important aspect: the calculation of a rotation matrix. In the future, the fuel carrier is controlled by the player. This means that the fuel carrier's orientation is always changing as the player races around collecting fuel cells. You have to account for this in the rendering code so that any changes by the player (such as turning left or right) are reflected in the game world. If fuel carrier orientation is not taken into account, you would have some very weird behavior for your game! At this point, you will add the support for this, but it won't be used until the next step: [FuelCell: What's My Motivation](#).

- Add the fuel carrier constants to the end of the `GameConstants` class, located in `GameConstants.cs`:

C#

```

//ship constants
public const float Velocity = 0.75f;
public const float TurnSpeed = 0.025f;
public const int MaxRange = 98;

```

As usual, the `FuelCarrier` data members are specific to the class. In this case, there is an orientation property, storing the current direction (in radians) that the fuel carrier is facing. This property is also used by the camera class to orientate along the same vector. The `MaxRange` member is used later to prevent the fuel carrier from driving off the playing field. This is something that would completely break the game play illusion.

As mentioned earlier, the methods are similar to the implementation code for the fuel cell and barrier classes. However, in the next part, you will add code that allows the player to drive the fuel carrier around the playing field. In fact, the fuel carrier has the singular honor of being the only moving part in the game!

Setting the Stage

It's time to shift our focus back to the main game class, `Game1.cs`. You're going to add member variables representing the new game objects you added: the fuel carrier, fuel cell, and various barriers. At this stage, we'll display a fuel cell, three barriers (each of a different type), and the fuel carrier on the playing field. Later in the development cycle, you'll add code that randomly generates and places the fuel cells and barriers.

However, before you start modifying this file, rename it `FuelCellGame.cs`. This follows the naming format of the other project files.

- In `FuelCellGame.cs`, after the declaration of the camera and ground variables, add the following code:

C#

```

Random random;
FuelCarrier fuelCarrier;
FuelCell[] fuelCells;
Barrier[] barriers;

```

- After the initialization of the camera and ground variables (located in the `Initialize` method), add the following code:

C#

```
//Initialize and place fuel cell
fuelCells = new FuelCell[1];
fuelCells[0] = new FuelCell();
fuelCells[0].LoadContent(Content, "Models/fuelcell");
fuelCells[0].Position = new Vector3(0, 0, 15);

//Initialize and place barriers
barriers = new Barrier[3];

barriers[0] = new Barrier();
barriers[0].LoadContent(Content, "Models/cube10uR");
barriers[0].Position = new Vector3(0, 0, 30);
barriers[1] = new Barrier();
barriers[1].LoadContent(Content, "Models/cylinder10uR");
barriers[1].Position = new Vector3(15, 0, 30);
barriers[2] = new Barrier();
barriers[2].LoadContent(Content, "Models/pyramid10uR");
barriers[2].Position = new Vector3(-15, 0, 30);

//Initialize and place fuel carrier
fuelCarrier = new FuelCarrier();
fuelCarrier.LoadContent(Content, "Models/fuelcarrier");
```

This code initializes all our new models and places them in front of the camera. The fuel cell is in the front row and the barriers are in a line behind it.

- Modify the existing `Draw` method by adding the following code after the `DrawTerrain` call:

C#

```
fuelCells[0].Draw(gameCamera.ViewMatrix,
    gameCamera.ProjectionMatrix);
foreach (Barrier barrier in barriers)
    barrier.Draw(gameCamera.ViewMatrix,
        gameCamera.ProjectionMatrix);
fuelCarrier.Draw(gameCamera.ViewMatrix,
    gameCamera.ProjectionMatrix);
```

Build and run the project and you will now see, in addition to the playing field, several cool things on the screen. You see some barriers, with a fuel cell slightly behind them, and a funny blue ovoid in the foreground. That is actually the fuel carrier. It's a (very) simple model, but it suits the purpose of the game. The next [step](#) implements user control of the game avatar.

See Also

Conceptual

[FuelCell](#)

Tasks

[How To: Render a Model](#)

FuelCell: What's My Motivation

Discusses the implementation of user control for the avatar (known as the fuel carrier).


The main points of this topic are:

- Receiving Player Input (illustrated by the following How Tos)
 - [How To: Detect Whether a Controller Button Is Pressed](#)
 - [How To: Detect Whether a Controller Button Has Been Pressed This Frame](#)
 - [How To: Detect Whether a Key Is Pressed](#)

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download FuelCell_4_Sample.zip](#).

 Note
You must download the above sample code in order to access the 3D models used in this tutorial step.

Overview

The goal for this part is the implementation of a control schema for the fuel carrier. With this schema, the player can use either the keyboard (A or D for left/right rotation and W and S for forward/backward movement) or a standard gamepad (using the left thumbstick for rotation and forward/backward movement). In addition, we'll check the `_maxRange` data member against the current position, and only allow movement that keeps the player on the playing field. This prevents the player from driving off the playing field.

GamePad States

In order to control the fuel carrier, you need input from a keyboard or gamepad. There are two approaches to getting input from the player: single-state and two-state. In a single-state approach, input is determined from a single snapshot of the controller, taken during execution of the `Update` method. Any actions that need to be taken by the game are initiated, thus enabling game play to move forward. This approach is demonstrated by [How To: Detect Whether a Controller Button Is Pressed](#).

However, when discrete input is required, the single-state approach doesn't solve the problem. For instance, suppose a game is designed to fire one bullet for every press of a key or button. If you use the single-state approach, multiple bullets are fired per key or button press. This happens because human reflexes are slower than the standard update cycle of the game. Even a very quick player is going to have a key or button pressed for at least a few update cycles (unless the game uses a fixed-step approach). In order to fire a single bullet every time a key or button is pressed, you must look for a current state where a specific key or button is released and a previous state where that same key or button was pressed. This condition is only satisfied at the instant when the key or button transitions from pressed to released.

Hence, the two-state approach: tracking the current state of the controller *and* the previous state of the controller. This allows the game to determine single occurrences of player action, such as a key or button press. With this approach, it doesn't matter how slow (or fast) the player's reflexes are. The input is only valid at that moment when the previous and current input states match the criteria determined by the input code. This approach is demonstrated by [How To: Detect Whether a Controller Button Has Been Pressed This Frame](#) and [How To: Detect Whether a Key Is Pressed](#).

It turns out that if you do any amount of XNA Game Studio game development, you often run into this controller update issue. The two-state approach is the best solution in most situations, and it is easy to implement. That same code also fulfills the needs of a single-state approach. In terms of keyboard and gamepad controls, FuelCell uses a two-state approach, but with a twist. The support for two-state input checking is in the code but FuelCell needs only check for current key and thumbstick states. Therefore, there is no need to check for a discrete event.

This code is an example of "future-proofing." If you were to add the capability to blow up a barrier with a missile, you would already have the necessary code to use the two-state approach.

Implementation begins in the FuelCellGame.cs file. Add the following code after the declaration of the `graphics` data member:

C#

```
KeyboardState lastKeyboardState = new KeyboardState();
```

```
KeyboardState currentKeyboardState = new KeyboardState();
GamePadState lastGamePadState = new GamePadState();
GamePadState currentGamePadState = new GamePadState();
```

Now, in the existing `Update` method, initialize the variables at the beginning of the method:

C#

```
lastKeyboardState = currentKeyboardState;
currentKeyboardState = Keyboard.GetState();
lastGamePadState = currentGamePadState;
currentGamePadState = GamePad.GetState(PlayerIndex.One);
```

You can now use either set of variables to determine exactly when a key or button is pressed or just use `currentGamePadState` if you only need the current position of the thumbstick. Your next step is to use this input to update the position and direction of the fuel carrier.

Updating the Fuel Carrier

Currently, the `FuelCarrier` class doesn't have a method for updating its position and heading. You'll add that method now and modify the main `Update` method to call it.

Staying within the `FuelCellGame.Update` function, replace this code:

C#

```
float rotation = 0.0f;
Vector3 position = Vector3.Zero;
gameCamera.Update(rotation, position,
    GraphicsDevice.Viewport.AspectRatio);
```

with the following:

C#

```
fuelCarrier.Update(currentGamePadState,
    currentKeyboardState, barriers);
float aspectRatio = graphics.GraphicsDevice.Viewport.AspectRatio;
gameCamera.Update(fuelCarrier.ForwardDirection,
    fuelCarrier.Position, aspectRatio);
```

The only difference is a call to the new update method, `FuelCarrier.Update`. This method takes as input the current keyboard and gamepad states and the `barriers` array. The method determines if the ship can move based on the current input and barrier locations. Ignore the `barriers` parameter for now; it is used in a later step. At this point, the function only prevents the vehicle from driving off the playing field. If an attempt is made to go over the playing field edge, the input is ignored.

Replace the code that checks for a Back button press with the following:

C#

```
// Allows the game to exit
if ((currentKeyboardState.IsKeyDown(Keys.Escape)) ||
    (currentGamePadState.Buttons.Back == ButtonState.Pressed))
    this.Exit();
```

The game now checks for both keyboard (ESC key) and gamepad input (Back button) when the player wishes to exit the game.

In the `GameObject.cs` file, locate your `FuelCarrier` class, and add the new update method:

C#

```
public void Update(GamePadState gamepadState,
    KeyboardState keyboardState, Barrier[] barriers)
{
    Vector3 futurePosition = Position;
    float turnAmount = 0;

    if (keyboardState.IsKeyDown(Keys.A))
```

```

    {
        turnAmount = 1;
    }
    else if(keyboardState.IsKeyDown(Keys.D))
    {
        turnAmount = -1;
    }
    else if(gamepadState.ThumbSticks.Left.X != 0)
    {
        turnAmount = -gamepadState.ThumbSticks.Left.X;
    }
    ForwardDirection += turnAmount * GameConstants.TurnSpeed;
    Matrix orientationMatrix = Matrix.CreateRotationY(ForwardDirection);

    Vector3 movement = Vector3.Zero;
    if (keyboardState.IsKeyDown(Keys.W))
    {
        movement.Z = 1;
    }
    else if(keyboardState.IsKeyDown(Keys.S))
    {
        movement.Z = -1;
    }
    else if (gamepadState.ThumbSticks.Left.Y != 0)
    {
        movement.Z = gamepadState.ThumbSticks.Left.Y;
    }

    Vector3 speed = Vector3.Transform(movement, orientationMatrix);
    speed *= GameConstants.Velocity;
    futurePosition = Position + speed;

    if (ValidateMovement(futurePosition, barriers))
    {
        Position = futurePosition;
    }
}

```

This method is important, so let's go through it in detail. First, the future position is set to the current position and the turn amount is set to 0. The amount of vehicle rotation is calculated, based on the current state of the A and D keys or the X-axis of the left thumbstick, and a rotation matrix is created. This matrix is later used to rotate the vehicle the proper amount in world coordinates. The second chunk of code calculates how far the ship moved in either a forward or backward direction, based on the current state of the W and S keys or the Y-axis of the same thumbstick. This distance is then transformed using the rotational matrix created earlier, and this result is multiplied by a constant velocity (`GameConstants.Velocity`). The final result is then added to the current position, resulting in a projected future position. Finally, this result is passed to a private method called `ValidateMovement`. If it is valid, the `_position` member is updated and control returns to the main `Game1.Update` method.

Programming Tip

You might be wondering why you couldn't just test the current position for validity instead of calculating a future position and testing that. The answer is that if you only tested the current position, it's already too late to prevent illegal movement. When the test is made, the vehicle has already had its position updated. Suppose that new position is illegal (past the edge of a boundary case). This causes further tests to fail, resulting in the vehicle "sticking" to the current position. Obviously, this is not optimal behavior for a player-controlled vehicle.

It is better to check the future position and prevent any illegal moves. This check allows the player to attempt something different (like backing up), and not get stuck because the player *currently* is in a legal position.

The remaining piece is the implementation of the `ValidateMovement` method. Add the following method after the `FuelCarrier.Update` method:

C#

```

private bool ValidateMovement(Vector3 futurePosition,
    Barrier[] barriers)
{

```

```
//Don't allow off-terrain driving
if ((Math.Abs(futurePosition.X) > MaxRange) ||
    (Math.Abs(futurePosition.Z) > MaxRange))
    return false;

return true;
}
```

Currently, this method only checks for the edge of the playing field. Any attempt to drive off the playing field is ignored.

After the usual drill of rebuilding the project and running it, drive the fuel carrier freely around the map. Test out the boundary code by driving to the edge of the playing field. You'll notice that you stop moving until you choose a new direction. The control schema implementation was pretty easy but, [coming up](#), the game really starts to come together... which requires a *lot* of coding!

See Also

Conceptual

[FuelCell](#)

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Button Has Been Pressed This Frame](#)

[How To: Detect Whether a Key Is Pressed](#)

FuelCell: What's My Line?

Demonstrates a simple, random technique for placing barriers and fuel cells on the playing field.

The main points of this topic are:

- Randomly Populating a Playing Field
- Designing Algorithms to Solve a Specific Problem

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download FuelCell_5_Sample.zip.](#)

Note

You must download the above sample code in order to access the 3D models used in this tutorial step.

Overview

The goal of this topic is the implementation of a good algorithm for placing a set number of barriers and fuel cells randomly throughout the playing field. The valid placement area (on a 100 x 100 unit playing field) is 90 x 90. A barrier centered on this limit does not overhang the playing field. In addition to an outer limit, you need an inner limit. This prevents a situation where the fuel carrier (always placed at the origin of the playing field: 0,0,0) is trapped by barriers. This still wouldn't be a problem at this stage because the game implements no collision detection, but in the finished product it would make for a frustrating game experience. Finally, another limit is used when placing a barrier. This is a minimum distance a new barrier must be from existing barriers. This prevents a collision with an existing barrier.

After testing with different values, 12 fuel cells and 40 barriers produced a challenging field. They are nicely cluttered with a good probability that many fuel cells are initially obscured by one or more barriers. You are encouraged to experiment with these numbers, but be warned that a higher total makes random placement of the barriers more difficult. The game could begin "thrashing," which means it is endlessly generating new random locations (to resolve a collision) only to find the new locations currently occupied.

A good example of unforeseen development problems was the coordinate system of the playing field. Because the playing field origin is at 0, 0, 0, the X and Z axes have both positive *and* negative values. This "automatically" divides the playing field into four quadrants because the sign can be either positive or negative values for the X and Z coordinates (the Y coordinate is clamped to 0). Therefore, randomly generating two positive numbers between the minimum and maximum distances populates only one quarter of the playing field. This is another bad gaming experience!

The solution used by FuelCell is to randomly assign positive and negative values to the randomly generated X and Z coordinates. This decently scatters the barriers around the playing field. However, due to the random nature of coordinate generation and limits imposed by a minimum and maximum, you'll notice that there is a bit of a "corridor" along the X and Z axes.

Pick Two: High Quality, Fast, and Cheap.

You might know (or at least heard) about the trilemma above. You're given three characteristics, but you can only choose two because of their interdependency. Any two of the characteristics negates the possibility of the third. For example, you can have something made cheaply and quickly, but it will be of poor quality.

Trilemmas can also be applied to software algorithms. A cheap, fast algorithm wastes a lot of resources compared to the result. On the other hand, a high quality algorithm that is fast is not cheap in terms of resources (in this case, development man hours).

The algorithm used by FuelCell is a fast, cheap algorithm. It took very little design time, and code, to implement, and it is as fast as other possible algorithms because the problem set is pretty small. However, the quality is very poor. It performs many unnecessary checks, it uses a brute-force approach when placing a new game object, and the randomness of the playing field is compromised in certain aspects (such as the axes corridors).

Initially, during early development of this game, a different algorithm was developed after observing the performance of the simple one used here. This algorithm was designed to prevent clustering and to generate an even spread of game objects throughout the playing field.

The solution was to treat the playing field as four separate 2D grids (or quadrants). This approach prevented the axes from fluctuating as the algorithm moved around the playing field grid. Each quadrant was populated individually using nested for loops: a loop for each axis. In addition, each quadrant was allocated a fourth of the total fuel cells and barriers. Once the buckets emptied, the algorithm moved onto the next quadrant. As the loops incremented, the algorithm automatically walked through each square in the quadrant (like iterating through a 2D array). As it walked through the possible placement areas, a random number was generated. If the random number matched a simple rule, it called another routine to place an object and zero out a weight counter. If there was no match, the algorithm placed nothing, incremented the weight counter, and moved on to the next grid location. If the weight counter reached a certain level, and pieces still remained for placement, it forced a piece placement. This ensured an even distribution throughout the quadrant.

The secondary function triggered when a piece needed to be placed. A random number (representing a percentile) was generated. Forty percent of the time a fuel cell was placed (if fuel cells remained in the quadrant allocation); otherwise, a barrier was placed (if barriers remained in the quadrant allocation). This matched the frequency of fuel cells to barriers and provided a general distribution. If the object being placed was a barrier, another random number determined the barrier type.

Finally, the algorithm verified that all game objects were placed before exiting the quadrant and that some type of object was placed, when required.

The end result was a consistently populated playing field that hardly ever had clusters or axes corridors. However, when this tutorial was written, the algorithm was overkill and introduced needless complexity to the main purpose: designing a 3D game. Therefore, the fast and cheap algorithm was used, making the code easier to understand.

Initializing the Random Number Generator

Since we are going to *randomly* populate the playing field with game objects, an obvious first step is to set up a random number generator. Let's keep it simple and declare a file-level random variable (in FuelCellGame.cs) that can be accessed by any FuelCellGame method.

After the gamepad state declarations, add the following:

C#

```
Random random;
```

Initialize the random number generator in the FuelCellGame constructor:

C#

```
random = new Random();
```

Modifying the LoadContent Method

In [FuelCell: What's My Motivation](#), we added some temporary code that created the FuelCell models on the playing field. You'll remove that code now, and initialize the arrays properly.

Remove the following from the Initialize method:

C#

```

//Initialize and place fuel cell
fuelCells = new FuelCell[1];
fuelCells[0] = new FuelCell();
fuelCells[0].LoadContent(Content, "Models/fuelcell");
fuelCells[0].Position = new Vector3(0, 0, 15);

//Initialize and place barriers
barriers = new Barrier[3];

barriers[0] = new Barrier();
barriers[0].LoadContent(Content, "Models/cube10uR");
barriers[0].Position = new Vector3(0, 0, 30);
barriers[1] = new Barrier();
barriers[1].LoadContent(Content, "Models/cylinder10uR");
barriers[1].Position = new Vector3(15, 0, 30);
barriers[2] = new Barrier();
barriers[2].LoadContent(Content, "Models/pyramid10uR");
barriers[2].Position = new Vector3(-15, 0, 30);

//Initialize and place fuel carrier
fuelCarrier = new FuelCarrier();
fuelCarrier.LoadContent(Content, "Models/fuelcarrier");

```

Add this code to the `LoadContent` method, after the loading of the ground game asset:

C#

```

//Initialize fuel cells
fuelCells = new FuelCell[GameConstants.NumFuelCells];
for (int index = 0; index < fuelCells.Length; index++)
{
    fuelCells[index] = new FuelCell();
    fuelCells[index].LoadContent(Content, "Models/fuelcell");
}

//Initialize barriers
barriers = new Barrier[GameConstants.NumBarriers];
int randomBarrier = random.Next(3);
string barrierName = null;

for (int index = 0; index < barriers.Length; index++)
{
    switch (randomBarrier)
    {
        case 0:
            barrierName = "Models/cube10uR";
            break;
        case 1:
            barrierName = "Models/cylinder10uR";
            break;
        case 2:
            barrierName = "Models/pyramid10uR";
            break;
    }
    barriers[index] = new Barrier();
    barriers[index].LoadContent(Content, barrierName);
    randomBarrier = random.Next(3);
}
PlaceFuelCellsAndBarriers();

//Initialize fuel carrier
fuelCarrier = new FuelCarrier();
fuelCarrier.LoadContent(Content, "Models/fuelcarrier");

```

Let's examine this code before moving on.

The first block initializes the array of fuel cells, loading each with the model for the fuel cell.

Barrier initialization is next. This code is a bit more complicated because there are three available barrier models. This looks like a job for the random number variable! A random number is generated and the corresponding model is loaded into the current barrier element using a **switch**> statement. The rest of the barrier object is initialized and a new random number is generated before moving to the next array element. After barrier initialization, the fuel cells and barriers are placed on the playing field with a call to `PlaceFuelCellsAndBarriers`. We'll discuss this function in detail later.

The final bit of code initializes and loads the model for the fuel carrier.

Fuel Cell and Barrier Initialization, Part 2

Now it is time to take a closer look at the `PlaceFuelCellsAndBarriers` method and its helper method, `GenerateRandomPosition`. Add the following code to the game (after the `LoadContent` method), and then we'll walk through it.

C#

```
private void PlaceFuelCellsAndBarriers()
{
    int min = GameConstants.MinDistance;
    int max = GameConstants.MaxDistance;
    Vector3 tempCenter;

    //place fuel cells
    foreach (FuelCell cell in fuelCells)
    {
        cell.Position = GenerateRandomPosition(min, max);
        tempCenter = cell.BoundingBox.Center;
        tempCenter.X = cell.Position.X;
        tempCenter.Z = cell.Position.Z;
        cell.BoundingBox = new BoundingBox(tempCenter,
            cell.BoundingBox.Width);
        cell.Retrieved = false;
    }

    //place barriers
    foreach (Barrier barrier in barriers)
    {
        barrier.Position = GenerateRandomPosition(min, max);
        tempCenter = barrier.BoundingBox.Center;
        tempCenter.X = barrier.Position.X;
        tempCenter.Z = barrier.Position.Z;
        barrier.BoundingBox = new BoundingBox(tempCenter,
            barrier.BoundingBox.Width);
    }
}

private Vector3 GenerateRandomPosition(int min, int max)
{
    int xValue, zValue;
    do
    {
        xValue = random.Next(min, max);
        zValue = random.Next(min, max);
        if (random.Next(100) % 2 == 0)
            xValue *= -1;
        if (random.Next(100) % 2 == 0)
            zValue *= -1;
    } while (IsOccupied(xValue, zValue));

    return new Vector3(xValue, 0, zValue);
}
```

It's not complicated, but it is also one of the bigger functions in the game. First, a few variables are declared, making the code more reader-friendly. The next part is a **foreach** loop that places the fuel cells. The algorithm is as follows:

1. Generate random values for the X and Z coordinates, verify that the new random location is not already occupied, and

update the fuel cell position with this new position.

Possible values are limited by the minimum and maximum placement values (defined in GameConstants.cs).

2. Initialize the bounding sphere property to the current fuel cell location.
3. Mark the fuel cell as unretrieved.

You follow the same process when you place the barriers.

The `GenerateRandomPosition` helper method makes up the remaining portion of newly added code. This method generates two random numbers. Another random number is generated and, depending on the result of the modulus operation (50% chance of negation), the X coordinate is negated. The same is done for the Z coordinate. The new position is then checked for existing occupants. If occupied, a new position is generated and the loop continues until a vacant location is found.

Let's add the new helper method, `IsOccupied`, to the project next.

Fuel Cell and Barrier Initialization, Part 3

Add this code after the `PlaceFuelCellsAndBarriers` method:

C#

```
private bool IsOccupied(int xValue, int zValue)
{
    foreach (GameObject currentObj in fuelCells)
    {
        if (((int)(MathHelper.Distance(
            xValue, currentObj.Position.X)) < 15) &&
            ((int)(MathHelper.Distance(
            zValue, currentObj.Position.Z)) < 15))
            return true;
    }

    foreach (GameObject currentObj in barriers)
    {
        if (((int)(MathHelper.Distance(
            xValue, currentObj.Position.X)) < 15) &&
            ((int)(MathHelper.Distance(
            zValue, currentObj.Position.Z)) < 15))
            return true;
    }
    return false;
}
```

This method uses the nifty `Distance` method when checking for collision with an existing game object (fuel cell or barrier). As you can see from the code, if the new object is closer than 15 units from an existing object, it is not placed. This can be modified, but keep in mind that the higher the distance, the more the placement method churns. It's that cheap, but fast effect again.

The final change to `FuelCellGame.cs` occurs in the `Draw` method. The new code draws all our wonderful fuel cells and barriers. Modify the contents of the `Draw` method to match the following:

C#

```
graphics.GraphicsDevice.Clear(Color.Black);

DrawTerrain(ground.Model);

foreach (FuelCell fuelCell in fuelCells)
    fuelCell.Draw(gameCamera.ViewMatrix,
        gameCamera.ProjectionMatrix);
foreach (Barrier barrier in barriers)
    barrier.Draw(gameCamera.ViewMatrix,
        gameCamera.ProjectionMatrix);
fuelCarrier.Draw(gameCamera.ViewMatrix,
    gameCamera.ProjectionMatrix);

base.Draw(gameTime);
```

New Game Constants

Before you can build and admire all your hard work, you need to add some constants to the GameConstants.cs file. Add the following code after any existing constants in the `GameConstants` class:

C#

```
//general
public const int MaxRangeTerrain = 98;
public const int NumBarriers = 40;
public const int NumFuelCells = 12;
public const int MinDistance = 10;
public const int MaxDistance = 90;
```

After a successful rebuild of the project, you now have fuel cells to find and barriers to avoid. Go ahead and take a spin around the new digs before moving on to adding a critical game feature: [collision detection](#).

See Also

Conceptual

[FuelCell](#)

FuelCell: "Ships" Passing in the Night

Discusses collision detection in a 3D game and demonstrates basic collision checking between the fuel carrier vehicle and the other game objects on the playing field.

The main points of this topic are:

- Collision Detection (illustrated by [How To: Detect Whether Two Models Collide](#))
- Custom Debugging Strategies

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download FuelCell_6_Sample.zip](#).

Note

You must download the above sample code in order to access the 3D models used in this tutorial step.

Out of My Way!

Currently, the game is not very challenging (and impossible to win). The player can drive through barriers and fuel cells without retrieving them. The only boundaries are the edges of the playing field, added in Step 4. It's time to implement the main feature of the game: collision detection.

Collision detection is a technique used in nearly all games, whether they are 2D or 3D. It simulates (to varying degrees) the interaction of real world objects. Without collision detection, objects in the game world could be seamlessly rendered within the space of the objects with which they collide. In most cases, this is unacceptable. The player expects to "collide" with objects in the game world and not merge with them. The desired behavior in FuelCell is to allow the player free motion around the playing field as long as he or she avoids the barriers. However, if the player attempts to drive through any part of a barrier, the player's current motion stops. Players must drive around the barriers, as they would in the real world. The behavior for fuel cells is a bit different. When a player "collides" with a fuel cell, the fuel cell object is marked as retrieved and no longer is drawn in the game world. This simulates the retrieval of the fuel cell. In the final step, the game display tracks these retrievals. The game ends when the player retrieves all the fuel cells.

FuelCell's implementation of collision detection is based on the [BoundingSphere](#) class and the following article: [How To: Detect Whether Two Models Collide](#). Collision detection with bounding spheres is a good approach for a variety of reasons:

- Bounding spheres (in addition to [bounding boxes](#) and [frustums](#)) are already available (courtesy of the XNA Framework) for use in your code.
- Bounding spheres do not have to be rotated if the enclosed object is rotated. This is not true when using bounding boxes or frustums. This saves some coding and reduces the complexity of the implementation.
- Bounding spheres are automatically generated for sub-meshes by the XNA Framework. Each mesh has a related bounding sphere, accessible with the [BoundingSphere](#) property of the [ModelMesh](#) object.

It's time to dive back into the code and take advantage of some of these features as we implement the collision detection.

Adding the Bounding Sphere Model

During development, it's incredibly useful to render your bounding spheres along with its related model. This makes debugging the collision detection code much easier when things don't seem to be working properly. FuelCell represents the bounding sphere as a spherical wire-frame model, whose radius is 1 unit. When rendering the bounding sphere of a model, it is easily scaled (using the radius of the model's bounding sphere) and placed by setting the bounding sphere's center to the current position of the model.

For example, if the radius of the bounding sphere of a mesh is 4, the bounding sphere model is scaled by 4, which matches the size of the original bounding sphere.

In order to use this technique, you need to add and initialize a spherical model with a radius of 1 unit. Luckily, the `sphere1uR.x` model is the model you need.

1. Add the `sphere1uR.x` and `White.png` files to the `Models/Content` directory of the FuelCell project.
2. Exclude the `White.png` file from the project.

3. In `FuelCellGame.cs`, declare a member variable (called `boundingSphere`) of type `GameObject`, after the declaration of the game camera.
4. In the existing `Initialize` method, initialize the new variable after initializing the game camera:

C#

```
boundingSphere = new GameObject();
```

5. In the existing `LoadContent` method, load the bounding sphere model after loading of the playing field model:

C#

```
boundingSphere.Model = Content.Load<Model>("Models/sphere1uR");
```

Once the model is loaded, you can use it to render the bounding spheres of your game objects. The only data you'll need to track is the size and position of each object's bounding sphere. For `FuelCell`, there are three types of bounding spheres: fuel carrier, fuel cell, and barrier. You can use the bounding sphere model for each type as long as you scale and position the model before rendering it. For simplicity's sake, the rendering code for a game object and its bounding sphere are kept together. Because the procedure for doing this is the same for both fuel cells and barriers, you will only walk through the fuel cell side of things. After that, we'll discuss the barrier implementation at a higher level.

Calculating the Boundary of the Fuel Cell Model

Do you remember the member variable you added previously to the `GameObject` class, `BoundingSphere`? This is the step that finally makes use of that variable, with the help of a bit of code. The first step involves calculating a decent approximation of the bounding sphere of the game object's model. We'll use the `CreateMerged` method. It's a quick and cheap way to calculate the bounding spheres of simple models. We'll start with the initial bounding sphere and, for every two spheres encountered in the model, we'll merge them. This continues until no more bounding spheres are left.

Note

For single mesh models, like those in `FuelCell`, this approach works perfectly. However, the approach begins to lose accuracy when applied to a complex model. This is why there are entire shelves of books devoted to solving this problem at your local bookstore.

Since we need the bounding sphere values from the model's sub-meshes, a good place for the initial calculation of this variable is in the `LoadContent` of the fuel cell model. Add the following code after any existing code:

C#

```
BoundingSphere = CalculateBoundingSphere();
```

This code makes a call to a helper function (`CalculateBoundingSphere`) that approximates the bounding sphere of the model, returning a new bounding sphere. The result is stored in the `BoundingSphere` variable. The Y-coordinate is set to 0. This causes the upper half of the bounding sphere to project out of the playing field. Since the widest part of the sphere is at the playing field level, it provides a good approximation when testing collision between fuel cells and the fuel carrier.

Add the helper method to the `GameObject` class, after any existing code:

C#

```
protected BoundingSphere CalculateBoundingSphere()
{
    BoundingSphere mergedSphere = new BoundingSphere();
    BoundingSphere[] boundingSpheres;
    int index = 0;
    int meshCount = Model.Meshes.Count;

    boundingSpheres = new BoundingSphere[meshCount];
    foreach (ModelMesh mesh in Model.Meshes)
    {
        boundingSpheres[index++] = mesh.BoundingSphere;
    }

    mergedSphere = boundingSpheres[0];
    if ((Model.Meshes.Count) > 1)
```

```

    {
        index = 1;
        do
        {
            mergedSphere = BoundingSphere.CreateMerged(mergedSphere,
                boundingSpheres[index]);
            index++;
        } while (index < Model.Meshes.Count);
    }
    mergedSphere.Center.Y = 0;
    return mergedSphere;
}

```

At this point, the bounding sphere has an approximate center (clamped to 0 on the Y axis) and radius. Remember the code you added in Step 5 that placed the fuel cells and barriers? After adding the bounding sphere calculation code, the approximated bounding sphere now has the proper coordinates of the related model. In the next step, you'll add code to render both fuel cells and their bounding spheres.

Drawing the Spheres

In the `Draw` function, replace the existing **foreach** loop that draws the fuel cells with the following:

C#

```

foreach (FuelCell fuelCell in fuelCells)
{
    if (!fuelCell.Retrieved)
    {
        fuelCell.Draw(gameCamera.ViewMatrix,
            gameCamera.ProjectionMatrix);
        GraphicsDevice.RenderState.FillMode =
            FillMode.WireFrame;
        fuelCell.DrawBoundingSphere(gameCamera.ViewMatrix,
            gameCamera.ProjectionMatrix, boundingSphere);
        GraphicsDevice.RenderState.FillMode =
            FillMode.Solid;
    }
}

```

What's new about this code? The most important changes are the graphics device state changes and the call to `DrawBoundingSphere`. The state changes are necessary to render the sphere model as a white wireframe model. We did this for two reasons: so the underlying model wouldn't be obscured, and the bounding sphere model would be easily visible. The switch to wireframe rendering is made by setting `RenderState.FillMode Property` to `FillMode.WireFrame`. Once the sphere model is drawn, the fill mode is set back to the default, `FillMode.Solid`. The second change is the call to a new draw function of the `GameObject` class, `DrawBoundingSphere`. Add the following code after the `CalculateBoundingSphere` method:

C#

```

internal void DrawBoundingSphere(Matrix view, Matrix projection,
    GameObject boundingSphereModel)
{
    Matrix scaleMatrix = Matrix.CreateScale(BoundingSphere.Radius);
    Matrix translateMatrix =
        Matrix.CreateTranslation(BoundingSphere.Center);
    Matrix worldMatrix = scaleMatrix * translateMatrix;

    foreach (ModelMesh mesh in boundingSphereModel.Model.Meshes)
    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.World = worldMatrix;
            effect.View = view;
            effect.Projection = projection;
        }
        mesh.Draw();
    }
}

```

Even though the two methods are similar, there are some important differences. First, the bounding sphere model is scaled using the radius of the fuel cell's bounding sphere instead of a constant factor. Second, the nice default lighting is turned off. This makes the model "pop" a bit more amidst the other models rendered with default lighting.

At this point, rebuild the game and start it. You'll notice that the fuel cells now have wireframe spheres surrounding them. You'll also notice that the spheres are pretty bad approximations of the fuel cell's boundaries. We'll get to that eventually. For now, let's add the code that updates and renders the bounding spheres for the fuel carrier and barriers.

Implementing the Remaining Bounding Spheres

First, let's update the barrier-related code since the barriers don't move, which results in less code to write.

- Modify the `LoadContent` method to calculate the initial bounding sphere.
- In the `Draw` method, modify the **foreach** loop (that draws the barriers) to draw the related bounding sphere.

Next up, the fuel carrier. It's the same procedure except for some extra code at the end to handle vehicle movement and fuel cell collision.

- Modify the `LoadContent` method to calculate the initial bounding sphere.
- Modify the `FuelCarrier.Update` method to update the bounding sphere location with the future position of the fuel carrier.
- In the `Draw` method, modify the fuel carrier draw code to also draw the related bounding sphere.

This completes the modifications to draw all the bounding spheres. If you rebuild and run the game, you'll now see lots of wire-frame spheres. But there is still more work (related to fuel carrier collision detection) to make these bounding spheres useful. At this point, the fuel carrier still ignores the fuel cell and barrier bounding spheres. Let's fix this now.

Collision Checking for the Fuel Carrier

There are two collision events that need implementation: collision with a fuel cell and collision with a barrier. If you remember, you already implemented the playing field edge collision code.

Fuel cell collision is important because that it is the only way the player can score points and win the game. You need to add code that checks for collisions between a fuel cell and the fuel carrier. If they collide, you need to mark the fuel cell as retrieved, update the retrieved fuel cells counter, and not draw the fuel cell in the future. This gives the impression that the fuel cell has been picked up by the fuel carrier.

Begin by modifying the `FuelCellGame.Update` method to check for collision between the fuel cell and the fuel carrier. Add the following code after the game camera update:

C#

```
foreach (FuelCell fuelCell in fuelCells)
    fuelCell.Update(fuelCarrier.BoundingSphere);
```

Add the `FuelCell.Update` method, called with the preceding code. Place it after the `Draw` method:

C#

```
internal void Update(BoundingSphere vehicleBoundingSphere)
{
    if (vehicleBoundingSphere.Intersects(this.BoundingSphere))
        this.Retrieved = true;
}
```

This checks for intersection between the bounding spheres of the carrier and the fuel cell. This is the essence of the procedure discussed in [How To: Detect Whether Two Models Collide](#).

Go ahead and remove the now-redundant check (located in `FuelCell.Draw`) to see if the fuel cell is retrieved before drawing. That check is now being made in the `FuelCellGame.Draw` method. This completes the code support for the first collision event.

The second event requires less code because, for the most part, the infrastructure already exists. The existing `ValidateMovement` method will be updated to check for barrier collision in addition to game field boundary collision.

Calculate the future position of the bounding sphere, based on the future position of the fuel carrier. Add the following code to the beginning of the `FuelCarrier.ValidateMovement` method:

C#

```
BoundingBox futureBoundingBox = BoundingBox;  
futureBoundingBox.Center.X = futurePosition.X;  
futureBoundingBox.Center.Z = futurePosition.Z;
```

In the same method, add the following code to the end of the current code but before the last **return** call:

C#

```
//Don't allow driving through a barrier  
if (CheckForBarrierCollision(futureBoundingBox, barriers))  
    return false;
```

Finally, add the new `CheckForBarrierCollision` method:

C#

```
private bool CheckForBarrierCollision(  
    BoundingBox vehicleBoundingBox, Barrier[] barriers)  
{  
    for (int curBarrier = 0; curBarrier < barriers.Length; curBarrier++)  
    {  
        if (vehicleBoundingBox.Intersects(  
            barriers[curBarrier].BoundingBox))  
            return true;  
    }  
    return false;  
}
```

Initial Bounding Sphere Results

Once you rebuild and run `FuelCell`, look at the size of the default bounding spheres. For each object, the bounding sphere is quite large. In some cases, the addition of bounding sphere volumes creates situations where the code prevents the vehicle from passing through barriers that are far enough apart for the vehicle itself. The collision detection, as it stands now, is pretty good considering that this is the result if you do nothing but use the default bounding sphere parameters. However, with a simple modification to the code we can improve the experience of the player. The modification involves applying another scaling factor to the bounding spheres of each game object. This allows realistic interaction between the vehicle and the game world.

The implementation requires scaling factors for each object type. A good place for these is in the `GameConstants.cs` file. Add the following declarations to the end of the `GameConstants` class:

C#

```
//bounding sphere scaling factors  
public const float FuelCarrierBoundingBoxFactor = .8f;  
public const float FuelCellBoundingBoxFactor = .5f;  
public const float BarrierBoundingBoxFactor = .7f;
```

There are factors for the fuel carrier, fuel cells, and barriers.

The second modification is to the `LoadContent` methods of the fuel carrier, fuel cell, and barrier classes. After the bounding sphere is first calculated, the radius is then modified by the appropriate factor. This shrinks the radius of the bounding sphere, creating a tighter fit with the object's visible boundary.

Modify the `FuelCell.LoadContent` method by adding the following code after the bounding sphere calculation:

C#

```
BoundingBox scaledSphere;  
scaledSphere = BoundingBox;  
scaledSphere.Radius *= GameConstants.FuelCellBoundingBoxFactor;  
BoundingBox =  
    new BoundingBox(scaledSphere.Center, scaledSphere.Radius);
```

Modify the `FuelCarrier.LoadContent` method by adding the following code after the bounding sphere calculation:

C#

```
BoundingBox scaledSphere;  
scaledSphere = BoundingBox;  
scaledSphere.Radius *=  
    GameConstants.FuelCarrierBoundingBoxFactor;  
BoundingBox =  
    new BoundingBox(scaledSphere.Center, scaledSphere.Radius);
```

Modify the `Barrier.LoadContent` method by adding the following code after the bounding sphere calculation:

C#

```
BoundingBox scaledSphere;  
scaledSphere = BoundingBox;  
scaledSphere.Radius *= GameConstants.BarrierBoundingBoxFactor;  
BoundingBox =  
    new BoundingBox(scaledSphere.Center, scaledSphere.Radius);
```

Rebuild and run the game. The bounding spheres are noticeably smaller, and they no longer enclose the game models. However, the important area (a band on the ground going around the game object) matches the outline of the model fairly accurately. At this point, the collision detection is good enough for our purposes. However, with a little effort, the placement and size of the bounding spheres could be modified even further to produce more accurate collision detection. Like most things in game development, the more time you spend on a feature, the better it gets (remember that trilemma we discussed earlier?). There's one more step before we can call it done. In the last [step](#), we'll add some new screens (for example, start, win or lose), game state tracking and HUD elements, and a few other items.

See Also

Conceptual

[FuelCell](#)

Tasks

[How To: Detect Whether Two Models Collide](#)

FuelCell: Finishing Touches

Discusses the final steps in the development of the FuelCell game.

The main points of this topic are:

- Displaying Game Status (illustrated by [How To: Draw Text](#))
- Game State Management
- Screen Transitioning Within a Game

The Complete Sample

The code in this tutorial illustrates the technique described in the text. A complete code sample for this tutorial is available for you to download, including full source code and any additional supporting files required by the sample.

[Download FuelCell_7_Sample.zip.](#)

Note

You must download the above sample code in order to access the 3D models used in this tutorial step.

Overview of Game State Management

As a game, FuelCell is nearly complete, but we are missing some important features. These features include:

- Game state management
- Start and win/loss screens
- Game status display (such as the current score and time remaining)

Game state management is the management of different states of the game through displaying different screens and/or by logic code added to the existing code. Common states for a basic game include: starting, playing, win/loss, and pause. Each of these states could have one or more related screens, such as an options screen or a gameplay screen.

FuelCell manages three states (starting, playing, won/lost) and three different screens:

- a starting screen with instructions on playing the game
- a gameplay screen displaying the fuel carrier and the playing field—the screen we have been working on from the beginning
- a screen that informs whether the player won or lost, and enables the player to start a new game or exit

The game state management code is confined to the FuelCellGame.cs file (with some constants defined in GameConstants.cs). Heavy modification of the `Update` and `Draw` methods is needed, and additional new code will be added in various areas of the file.

However, before we start that work, remove the code that renders the bounding spheres. It served its purpose and only gets in the way now. In the `Draw` method, comment out (or remove) all code that sets the wire frame rendering mode, and calls the `DrawBoundingSphere` method. Rebuild the game and run it. Without all the wireframe spheres, it looks like a real game.

Note

If this were a real game in development, you would either remove these calls completely or enclose them in `DEBUG` conditionals. In addition, you could remove or conditionalize the code around initializing and loading the bounding sphere model (`BoundingSphere`). However, you still need the remaining code initializing and calculating the bounding spheres for the various game objects. Without it, our collision detection disappears.

For our purposes, a simple commenting out suffices.

Anybody Got the Time?

One of the goals of FuelCell is to collect 12 fuel cells. It's not a very difficult task if the player has all the time in the world. Let's fix that now by adding a time limit to the game. The player starts with a set amount of time. In this case, we'll use 30 seconds. If the player collects all 12 fuel cells before time runs out, the player wins the game. However, if time runs out and there are still fuel cells on the playing field, the player loses. The winning and losing conditions are handled by the game state management code that you'll add later. For now, let's implement the time limit feature.

In `FuelCellGame.cs`, declare three variables (of type `TimeSpan`) for the various times to be tracked. You'll also need a variable for tracking the number of retrieved fuel cells (of type `int`).

C#

```
int retrievedFuelCells;
TimeSpan startTime, roundTimer, roundTime;
```

In the `FuelCellGame` constructor, add initialization code for the starting and round times. This code sets the start and game time variables to 0. Zero indicates the game hasn't started. The round time variable is then set to 30.25 seconds. The .25 seconds is to give enough time to flash 30 on the screen before it begins counting down.

C#

```
roundTime = GameConstants.RoundTime;
```

While you are in here, add the following code to specify a resolution of 853x480. This resolution causes output from any Xbox console to look good, regardless of the widescreen capabilities of the output device.

C#

```
graphics.PreferredBackBufferWidth = 853;
graphics.PreferredBackBufferHeight = 480;
```

In the `Update` method, add code that sets the time variables to their proper values when the player presses the Start button on the gamepad. A good place for this code is after the code that checks for pressing of the Exit button:

```
if (currentGamePadState.Buttons.Start == ButtonState.Pressed)
{
    roundTimer = roundTime;
}
```

This code sets the current and start times to the current total game time, and then calculates the end time. Now that the time is set, you'll need to track the amount of elapsed game time.

Staying in the same method, add the following code after the fuel cell update code:

```
roundTimer -= gameTime.ElapsedGameTime;
```

Gimme a T!

Next, you'll implement a simple HUD that displays the remaining time and the number of retrieved fuel cells during gameplay. We'll accomplish this by using the technique described by [How To: Draw Text](#).

The first item is a spritefont for the screen text. This is a new content type for the game, so add a new directory to the Content directory called `Fonts`, and then add a new `Spritefont` item to that directory called `StatsFont`.

In the `GameConstants.cs` file, add the following strings:

C#

```
public const string StrTimeRemaining = "Time Remaining: ";
public const string StrCellsFound = "Fuel Cells Retrieved: ";
public const string StrGameWon = "Game Won !";
public const string StrGameLost = "Game Lost !";
public const string StrPlayAgain =
    "Press Enter/Start to play again or Esc/Back to quit";
public const string StrInstructions1 =
    "Retrieve all Fuel Cells before time runs out.";
public const string StrInstructions2 =
    "Control ship using keyboard (A, D, W, S) or the left thumbstick.";
```

These strings are used in various screens to display information on how to play, the current time remaining, and other important stuff. You'll also need a sprite batch and a sprite font variable when drawing the game text. Add these in the `FuelCellGame.cs` file after the `random` variable declaration:

C#

```
SpriteBatch spriteBatch;  
SpriteFont statsFont;
```

In the `LoadContent` method, initialize the sprite batch object, and load the new sprite font with the following code:

C#

```
spriteBatch = new SpriteBatch(GraphicsDevice);  
statsFont = Content.Load<SpriteFont>("Fonts/StatsFont");
```

You'll use a single method, called from `Draw`, to draw all the game statistics on the gameplay screen. In the `FuelCellGame.cs` file, add the following method after the `DrawTerrain` method:

C#

```
private void DrawStats()  
{  
    float xOffsetText, yOffsetText;  
    string str1 = GameConstants.StrTimeRemaining;  
    string str2 =  
        GameConstants.StrCellsFound + retrievedFuelCells.ToString() +  
        " of " + GameConstants.NumFuelCells.ToString();  
    Rectangle rectSafeArea;  
  
    str1 += (roundTimer.Seconds).ToString();  
  
    //Calculate str1 position  
    rectSafeArea = GraphicsDevice.Viewport.TitleSafeArea;  
  
    xOffsetText = rectSafeArea.X;  
    yOffsetText = rectSafeArea.Y;  
  
    Vector2 strSize = statsFont.MeasureString(str1);  
    Vector2 strPosition =  
        new Vector2((int)xOffsetText + 10, (int)yOffsetText);  
  
    spriteBatch.Begin();  
    spriteBatch.DrawString(statsFont, str1, strPosition, Color.White);  
    strPosition.Y += strSize.Y;  
    spriteBatch.DrawString(statsFont, str2, strPosition, Color.White);  
    spriteBatch.End();  
  
    //re-enable depth buffer after sprite batch disablement  
    GraphicsDevice.RenderState.DepthBufferEnable = true;  
    GraphicsDevice.RenderState.AlphaBlendEnable = false;  
    GraphicsDevice.RenderState.AlphaTestEnable = false;  
    GraphicsDevice.SamplerStates[0].AddressU = TextureAddressMode.Wrap;  
    GraphicsDevice.SamplerStates[0].AddressV = TextureAddressMode.Wrap;  
}
```

Let's look at this code block more closely as there's a fair bit of code here.

The first part sets up the locations where the game statistic strings are drawn. A major part of this is the retrieval of the title safe area of the viewport. This is later used to properly place the various text strings. A nice feature of the `TitleSafeArea` property is that, depending on the platform, it automatically adjusts the area returned. The area is smaller if the game targets the Xbox 360 platform. Rendering within this area prevents any sprite drawing in areas of the screen that could get cut off by a TV display. The next part composes the strings with the current game info, and then draws them on the gameplay screen. However, the last bit of code deserves more examination.

You must reset certain properties of the graphics device whenever you combine sprite drawing with 3D rendering, as `FuelCell` does. These properties, related to alpha blending and the depth buffer, are set to different values when a sprite batch is used. If the properties are not reset to the default settings, weird rendering issues could suddenly appear in your game. That is why the final code modifies some `RenderState` and `SampleStates` properties of the graphics device.

You'll need to call the `DrawStats` method from the existing `Draw` method. Add the following code after the call to

```
FuelCarrier.Draw.
```

```
DrawStats();
```

After a rebuild, the time and remaining number of fuel cells are displayed in the upper-left corner. Of course, the values are completely wrong, but we'll be fixing those next.

The State of Things

The remaining code you will add implements the game state management of FuelCell. As mentioned previously, one of the main duties of game state management is to determine the correct game screen to display. In addition, game state management can also monitor the player's progress toward his or her goals. For FuelCell, the goals are to retrieve all fuel cells before time runs out. If the player succeeds, we need to display a Game Won screen or, conversely, a Game Lost screen. In addition, we'll display a splash screen on startup of the game containing instructions for starting a new game and playing it.

The state of the game changes depending on player input and events in the game. This makes the `Update` method an ideal place to monitor and control the varying states. In addition, the `Draw` method will be modified to properly reflect the current stage the game is in by drawing the proper screen.

Let's begin by declaring a new enumeration in the FuelCellGame.cs file that lists the different game states being tracked. Add the following code before the `FuelCellGame` class declaration:

C#

```
public enum GameState { Loading, Running, Won, Lost }
```

Now add a variable (of type `GameState`) after the declaration of the `SpriteBatch` and `SpriteFont` variables.

C#

```
GameState currentGameState = GameState.Loading;
```

We're going to implement the game state component in stages, with the first stage being the Loading state. The first part, setting the current game state to Loading, is already done.

Implementing the Loading Game State

Before the player starts a new game of FuelCell, the player will need some basic instructions for controlling the vehicle and the win conditions. We'll deliver these instructions in a splash screen, which will display when the FuelCell application begins. This screen is only displayed when the game state is in the `Loading` stage.

Modify the following code:

```
        if ((lastKeyboardState.IsKeyDown(Keys.Enter) && (lastKeyboardState.IsKeyUp(Keys.Enter))) ||
            currentGamePadState.Buttons.Start == ButtonState.Pressed)
        {
            roundTimer = roundTime;
        }
```

to match the following:

C#

```
if (currentGameState == GameState.Loading)
{
    if ((lastKeyboardState.IsKeyDown(Keys.Enter) &&
        (currentKeyboardState.IsKeyUp(Keys.Enter))) ||
        currentGamePadState.Buttons.Start == ButtonState.Pressed)
    {
        roundTimer = roundTime;
        currentGameState = GameState.Running;
    }
}
```

The modifications now check the game state when the player presses the Enter key or Start button, and advances the game state if the current state is `GameState.Loading`.

Display of the splash screen is handled in the `Draw` method. Since the game state is based on the current value of an enumeration, we'll draw the proper screen based on the current value of `currentGameState`.

In the `Draw` method, replace the existing code between the `Clear` call and the base `Draw` call with the following code:

```
switch (currentGameState)
{
    case GameState.Loading:
        DrawSplashScreen();
        break;
};
```

Before the player starts the game, the only thing displayed should be the splash screen. The helper method, `DrawSplashScreen`, clears the screen to steel blue, and then displays some informational text. Add that method after the existing `DrawTerrain` method:

C#

```
private void DrawSplashScreen()
{
    float xOffsetText, yOffsetText;
    Vector2 viewportSize = new Vector2(GraphicsDevice.Viewport.Width,
        GraphicsDevice.Viewport.Height);
    Vector2 strCenter;

    graphics.GraphicsDevice.Clear(Color.SteelBlue);

    xOffsetText = yOffsetText = 0;
    Vector2 strInstructionsSize =
        statsFont.MeasureString(GameConstants.StrInstructions1);
    Vector2 strPosition;
    strCenter = new Vector2(strInstructionsSize.X / 2,
        strInstructionsSize.Y / 2);

    yOffsetText = (viewportSize.Y / 2 - strCenter.Y);
    xOffsetText = (viewportSize.X / 2 - strCenter.X);
    strPosition = new Vector2((int)xOffsetText, (int)yOffsetText);

    spriteBatch.Begin();
    spriteBatch.DrawString(statsFont, GameConstants.StrInstructions1,
        strPosition, Color.White);

    strInstructionsSize =
        statsFont.MeasureString(GameConstants.StrInstructions2);
    strCenter = new Vector2(strInstructionsSize.X / 2,
        strInstructionsSize.Y / 2);
    yOffsetText =
        (viewportSize.Y / 2 - strCenter.Y) + statsFont.LineSpacing;
    xOffsetText = (viewportSize.X / 2 - strCenter.X);
    strPosition = new Vector2((int)xOffsetText, (int)yOffsetText);

    spriteBatch.DrawString(statsFont, GameConstants.StrInstructions2,
        strPosition, Color.LightGray);
    spriteBatch.End();

    //re-enable depth buffer after sprite batch disablement
    GraphicsDevice.RenderState.DepthBufferEnable = true;
    GraphicsDevice.RenderState.AlphaBlendEnable = false;
    GraphicsDevice.RenderState.AlphaTestEnable = false;
    GraphicsDevice.SamplerStates[0].AddressU = TextureAddressMode.Wrap;
    GraphicsDevice.SamplerStates[0].AddressV = TextureAddressMode.Wrap;
}
```

That completes the splash screen implementation.

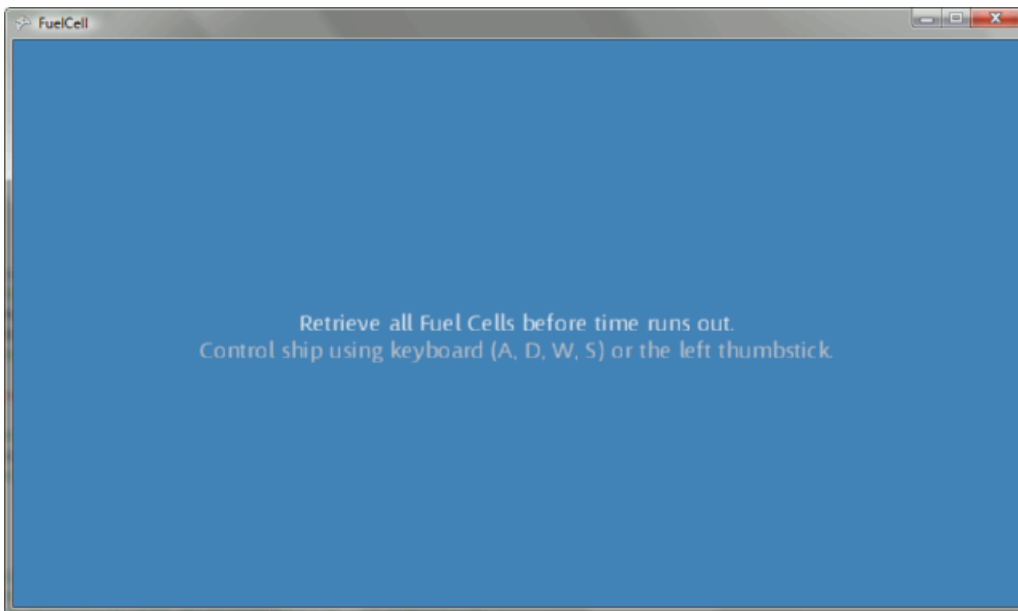


Figure 1. Splash Screen for FuelCell

Implementing the Running (Gameplay) Screen

The next game state you'll implement is the gameplay screen (the screen that you developed in Steps 3-6). This screen is reached after the player presses the Enter key or the Start button from the splash screen. You'll add code to the `Update` and `Draw` methods to draw this screen at the appropriate time.

In the `Update` method, replace the following code:

```
fuelCarrier.Update(currentGamePadState, currentKeyboardState, barriers);
float aspectRatio = graphics.GraphicsDevice.Viewport.AspectRatio;
gameCamera.Update(fuelCarrier.ForwardDirection, fuelCarrier.Position, aspectR
atio);

foreach (FuelCell fuelCell in fuelCells)
fuelCell.Update(fuelCarrier.BoundingBox);
roundTimer -= gameTime.ElapsedGameTime;
```

with the following code:

C#

```
if ((currentGameState == GameState.Running))
{
    fuelCarrier.Update(currentGamePadState,
        currentKeyboardState, barriers);
    gameCamera.Update(fuelCarrier.ForwardDirection,
        fuelCarrier.Position, aspectRatio);
    retrievedFuelCells = 0;
    foreach (FuelCell fuelCell in fuelCells)
    {
        fuelCell.Update(fuelCarrier.BoundingBox);
        if (fuelCell.Retrieved)
        {
            retrievedFuelCells++;
        }
    }
    if (retrievedFuelCells == GameConstants.NumFuelCells)
    {
        currentGameState = GameState.Won;
    }
    roundTimer -= gameTime.ElapsedGameTime;
    if ((roundTimer < TimeSpan.Zero) &&
        (retrievedFuelCells != GameConstants.NumFuelCells))
    {
        currentGameState = GameState.Lost;
    }
}
```

```
}
```

In the `Draw` method, add the following code after the existing game state check:

C#

```
case GameState.Running:  
    DrawGameplayScreen();  
    break;
```

And then, after the `DrawSplashScreen` method, add the following code:

C#

```
private void DrawGameplayScreen()  
{  
    DrawTerrain(ground.Model);  
    foreach (FuelCell fuelCell in fuelCells)  
    {  
        if (!fuelCell.Retrieved)  
        {  
            fuelCell.Draw(gameCamera.ViewMatrix,  
                gameCamera.ProjectionMatrix);  
        }  
    }  
    foreach (Barrier barrier in barriers)  
    {  
        barrier.Draw(gameCamera.ViewMatrix,  
            gameCamera.ProjectionMatrix);  
    }  
  
    fuelCarrier.Draw(gameCamera.ViewMatrix,  
        gameCamera.ProjectionMatrix);  
    DrawStats();  
}
```

At this point, the game displays both the splash and gameplay screens. Only the win/loss screen remains.

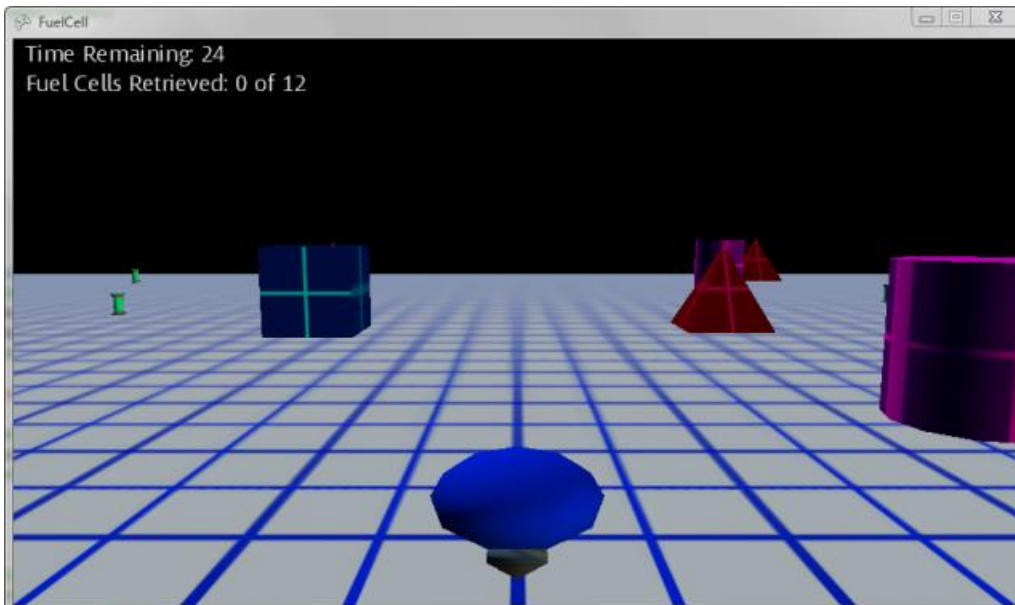


Figure 2. Gameplay Screen for FuelCell

Implementing the Game Won/Game Lost Screens

The win/loss screens are similar, differing only in the initial text. From either screen, the player can choose to play another game or exit FuelCell. For this reason, you'll implement both screens using a single function. That function, `DrawWinOrLoss`, accepts a string parameter with the appropriate message.

First, let's modify the `Update` method to check for player input after a game. Add the following code after the check for `GameState.Running`:

C#

```
if ((currentGameState == GameState.Won) ||
    (currentGameState == GameState.Lost))
{
    // Reset the world for a new game
    if ((lastKeyboardState.IsKeyDown(Keys.Enter) &&
        (currentKeyboardState.IsKeyUp(Keys.Enter))) ||
        currentGamePadState.Buttons.Start == ButtonState.Pressed)
        ResetGame(gameTime, aspectRatio);
}
```

There's nothing of note in this code except for the call to the `Reset` method. It "resets" the world, allowing a new game to be played. Let's add that method now.

In `FuelCellGame.cs`, add the following method:

C#

```
private void ResetGame(GameTime gameTime, float aspectRatio)
{
    fuelCarrier.Reset();
    gameCamera.Update(fuelCarrier.ForwardDirection,
        fuelCarrier.Position, aspectRatio);
    InitializeGameField();

    retrievedFuelCells = 0;
    startTime = gameTime.TotalGameTime;
    roundTimer = roundTime;
    currentGameState = GameState.Running;
}
```

In the `GameObject.cs` file, after the `LoadContent` method, add the following method to the `FuelCarrier` class:

C#

```
internal void Reset()
{
    Position = Vector3.Zero;
    ForwardDirection = 0f;
}
```

This returns the fuel carrier to its original position and orientation.

The final code you'll add is `InitializeGameField`. Place this code after the `ResetGame` method (located in `FuelCellGame.cs`):

C#

```
private void InitializeGameField()
{
    //Initialize barriers
    barriers = new Barrier[GameConstants.NumBarriers];
    int randomBarrier = random.Next(3);
    string barrierName = null;

    for (int index = 0; index < GameConstants.NumBarriers; index++)
    {
        switch (randomBarrier)
        {
            case 0:
                barrierName = "Models/cube10uR";
                break;
            case 1:
                barrierName = "Models/cylinder10uR";
                break;
            case 2:
                barrierName = "Models/pyramid10uR";
                break;
        }
    }
}
```



```

    }
    barriers[index] = new Barrier();
    barriers[index].LoadContent(Content, barrierName);
    randomBarrier = random.Next(3);
}
PlaceFuelCellsAndBarriers();
}

```

This method completes the remaining tasks for a new game:

- Generates new positions for all fuel cells and barriers.
- Sets all fuel cells to a non-retrieved state and zeros out the number of retrieved fuel cells.

The final bit of code updates the `Draw` method to draw the winning or losing screen. Add the following code after the check for the Running game state in `Draw`:

C#

```

case GameState.Won:
    DrawWinOrLossScreen(GameConstants.StrGameWon);
    break;
case GameState.Lost:
    DrawWinOrLossScreen(GameConstants.StrGameLost);
    break;

```

Now implement the `DrawWinOrLossScreen` method, after the existing `DrawSplashScreen` method:

C#

```

private void DrawWinOrLossScreen(string gameResult)
{
    float xOffsetText, yOffsetText;
    Vector2 viewportSize = new Vector2(GraphicsDevice.Viewport.Width,
        GraphicsDevice.Viewport.Height);
    Vector2 strCenter;

    xOffsetText = yOffsetText = 0;
    Vector2 strResult = statsFont.MeasureString(gameResult);
    Vector2 strPlayAgainSize =
        statsFont.MeasureString(GameConstants.StrPlayAgain);
    Vector2 strPosition;
    strCenter = new Vector2(strResult.X / 2, strResult.Y / 2);

    yOffsetText = (viewportSize.Y / 2 - strCenter.Y);
    xOffsetText = (viewportSize.X / 2 - strCenter.X);
    strPosition = new Vector2((int)xOffsetText, (int)yOffsetText);

    spriteBatch.Begin();
    spriteBatch.DrawString(statsFont, gameResult,
        strPosition, Color.Red);

    strCenter =
        new Vector2(strPlayAgainSize.X / 2, strPlayAgainSize.Y / 2);
    yOffsetText = (viewportSize.Y / 2 - strCenter.Y) +
        (float)statsFont.LineSpacing;
    xOffsetText = (viewportSize.X / 2 - strCenter.X);
    strPosition = new Vector2((int)xOffsetText, (int)yOffsetText);
    spriteBatch.DrawString(statsFont, GameConstants.StrPlayAgain,
        strPosition, Color.AntiqueWhite);

    spriteBatch.End();

    //re-enable depth buffer after sprite batch disablement
    GraphicsDevice.RenderState.DepthBufferEnable = true;
    GraphicsDevice.RenderState.AlphaBlendEnable = false;
    GraphicsDevice.RenderState.AlphaTestEnable = false;
    GraphicsDevice.SamplerStates[0].AddressU = TextureAddressMode.Wrap;
    GraphicsDevice.SamplerStates[0].AddressV = TextureAddressMode.Wrap;

```

```
}
```

Since the Win and Loss screens differ only in the message, we can cheat a bit and just use the same function, passing in the game result.

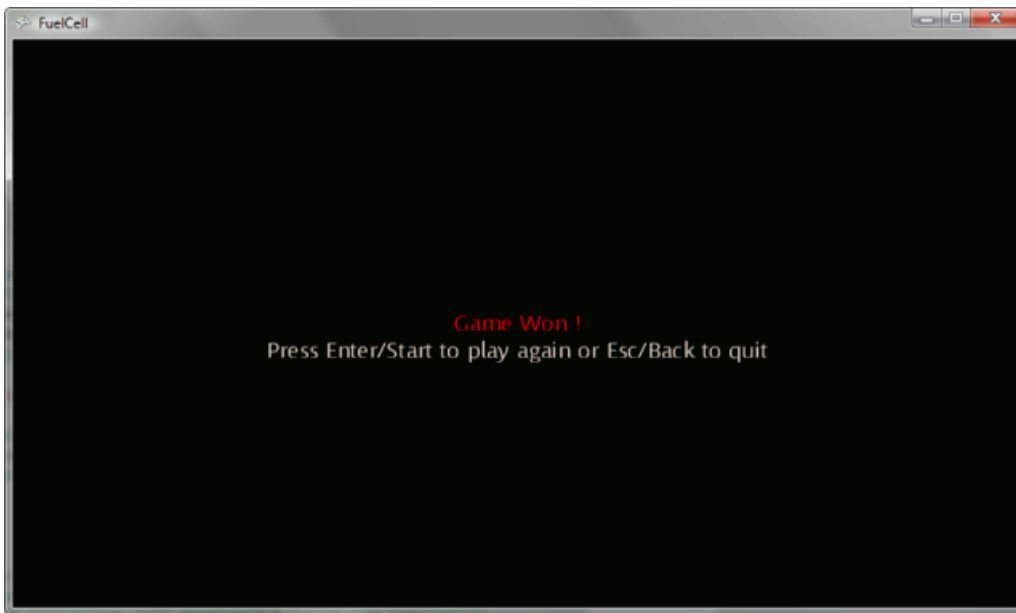


Figure 3. FuelCell Congratulatory Screen

That's a Wrap!

After the (now) traditional rebuild and play session, you can now enjoy the introductory splash screen, a challenging game of retrieve the fuel cells, and (hopefully) a congratulatory winning screen.

At this point, the FuelCell game is complete. Even though you wrote a large amount of code, the possibilities for expansion are endless. For example:

- Multiplayer - invite a competitor and whomever collects the most fuel cells wins!
- Powerups - Add another object, or two, that gives the player a boost of speed for 5 seconds or a "ghost" ability that ignores those pesky barriers!
- Advanced lighting and effects - improve the default lighting with a glow effect or motion blur.
- Sound - add the Whoosh! of fuel carrier acceleration or the Bonk! of barrier collision.

If you need more ideas for expanding FuelCell or want to interact with a like-minded community of fellow game developers, check out [XNA Creators Club Online Web site](#). It's an excellent community-driven site that has active forums and helpful people who are designing new and exciting games.

Good luck in your game development future and, above all, have fun!

See Also

Conceptual

[FuelCell](#)

Tasks

[How To: Draw a Sprite](#)

[How To: Draw Text](#)

XNA Framework Class Library

The XNA Framework class library is a library of classes, interfaces, and value types that are included in XNA Game Studio. This library provides access to XNA Framework functionality and is designed to be the foundation on which XNA Game Studio applications, components, and controls are built.

Namespaces

[Microsoft.Xna.Framework](#)

Provides commonly needed game classes such as timers and game loops.

[Microsoft.Xna.Framework.Audio](#)

Contains low-level application programming interface (API) methods that can load and manipulate XACT-created project and content files to play audio.

[Microsoft.Xna.Framework.Content](#)

Contains the run-time components of the Content Pipeline.

[Microsoft.Xna.Framework.Design](#)

Provides a unified way of converting types of values to other types.

[Microsoft.Xna.Framework.GamerServices](#)

Contains classes that implement various services related to gamers. These services communicate directly with the gamer, the gamer's data, or otherwise reflect choices the gamer makes. Gamer services include input device and profile data APIs.

[Microsoft.Xna.Framework.Graphics](#)

Contains low-level application programming interface (API) methods that take advantage of hardware acceleration capabilities to display 3D objects.

[Microsoft.Xna.Framework.Graphics.PackedVector](#)

Represents data types with components that are not multiples of 8 bits.

[Microsoft.Xna.Framework.Input](#)

Contains classes to receive input from keyboard, mouse, and Xbox 360 Controller devices.

[Microsoft.Xna.Framework.Media](#)

Contains classes to enumerate, play, and view songs, albums, playlists, and pictures.

[Microsoft.Xna.Framework.Net](#)

Contains classes that implement support for Xbox LIVE, multiplayer, and networking for XNA Framework games.




















[Microsoft.Xna.Framework.Storage](#)

Contains classes that allow reading and writing of files.





Microsoft.Xna.Framework Namespace

Provides commonly needed game classes such as timers and game loops.












Classes

Name	Description
 BoundingFrustum	Defines a frustum and helps determine whether forms intersect with it.
 Curve	Stores an arbitrary collection of 2D CurveKey points, and provides methods for evaluating features of the curve they define.
 CurveKey	Represents a point in a multi-point curve.
 CurveKeyCollection	Contains the CurveKeys making up a Curve .
 DrawableGameComponent	A game component that is notified when it needs to draw itself.
 FrameworkDispatcher	Implements the Windows-specific portion of a FrameworkDispatcher class.
 Game	Provides basic graphics device initialization, game logic, and rendering code.
 GameComponent	Base class for all XNA Framework game components.
 GameComponentCollection	A collection of game components.
 GameComponentCollectionEventArgs	Arguments used with events from the GameComponentCollection .
 GameServiceContainer	A collection of game services.
 GameTime	Snapshot of the game timing state expressed in values that can be used by variable-step (real time) or fixed-step (game time) games.
 GameWindow	The system window associated with a Game .
 GraphicsDeviceInformation	Holds the settings for creating a graphics device on Windows.
 GraphicsDeviceManager	Handles the configuration and management of the graphics device.
 MathHelper	Contains commonly used precalculated values.
 NoSuitableGraphicsDeviceException	Thrown when no available graphics device fits the given device preferences.
 PowerStatus	Contains system power information, including battery life status and power line status.
 PreparingDeviceSettingsEventArgs	Arguments for the GraphicsDeviceManager.PreparingDeviceSettings event.










Interfaces

Name	Description
 IDrawable	Defines the interface for a drawable game component.
 IGameComponent	Defines an interface for game components.
 IGraphicsDeviceManager	Defines the interface for an object that manages a GraphicsDevice .
 IUpdateable	Defines an interface for a game component that should be updated in Game.Update .

Structures

Name	Description
 BoundingBox	Defines an axis-aligned box-shaped 3D volume.
 BoundingSphere	Defines a sphere.
 Matrix	Defines a matrix.
 Plane	Defines a plane.
 Point	Defines a point in 2D space.
 Quaternion	Defines a four-dimensional vector (x,y,z,w), which is used to efficiently rotate an object about the (x, y, z) vector by the angle theta, where $w = \cos(\text{theta}/2)$.
 Ray	Defines a ray.
 Rectangle	Defines a rectangle.
 Vector2	Defines a vector with two components.
 Vector3	Defines a vector with three components.
 Vector4	Defines a vector with four components.

Enumerations

Name	Description
 BatteryChargeStatus	Indicates battery charge status.
 ContainmentType	Indicates the extent to which bounding volumes intersect or contain one another.
 CurveContinuity	Defines the continuity of CurveKeys on a Curve .
 CurveLoopType	Defines how the value of a Curve will be determined for positions before the first point on the Curve or after the last point on the Curve .
 CurveTangent	Specifies different tangent types to be calculated for CurveKey points in a Curve .
 PlaneIntersectionType	Describes the intersection between a plane and a bounding volume.
 PlayerIndex	Specifies the game controller associated with a player.
 PowerLineStatus	Indicates the status of the system powerline.
 TargetPlatform	Defines the target platform to be used when compiling content.

See Also

Tasks

- [How To: Allow the Player to Resize a Game Window](#)
- [How To: Pause a Game](#)
- [How To: Exit a Game](#)
- [How To: Display a Game in Full-Screen Mode](#)
- [How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)
- [How To: Make a Game Time Out](#)
- [How To: Make a Game Use a Variable Time Step](#)
- [How To: Transform a Point with a Matrix](#)
- [How To: Rotate and Move a Camera](#)
- [How To: Make a First-Person Camera](#)
- [How To: Make a Third-Person Camera](#)
- [How To: Script the Camera to Follow a Curve](#)
- [How To: Position the Camera to View All Objects in a Scene](#)
- [How To: Detect Whether Two Models Collide](#)
- [How To: Detect Whether a User Clicked a 3D Object](#)

Concepts

- [Application Model Overview](#)
- [Math Overview](#)
- [Math Content Catalog at XNA Creators Club Online](#)

BatteryChargeStatus Enumeration

Indicates battery charge status.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum BatteryChargeStatus
```

Members

Member name	Description
Charging	Indicates a battery is charging.
Critical	Indicates a critically low level of battery charge.
High	Indicates a high level of battery charge.
Low	Indicates a low level of battery charge.
NoSystemBattery	Indicates that no battery is present.
Unknown	Indicates an unknown battery condition.

Remarks

The [PowerStatus.BatteryChargeStatus](#) property returns a **BatteryChargeStatus** value.

See Also

Reference

[PowerStatus.BatteryChargeStatus Property](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox Structure

Defines an axis-aligned box-shaped 3D volume.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.BoundingBoxConverter)")]  
[SerializableAttribute]  
public struct BoundingBox : IEquatable<BoundingBox>
```

Remarks

The [BoundingBox Structure](#) represents the space occupied by a box. The bounding box class is axis aligned. Each face of the bounding box is perpendicular to the x-axis, the y-axis, or the z-axis.

There are several benefits of using the bounding box for collision detection.

- The bounding box class fits rectangular shapes aligned with the axis very well. Compared to the bounding sphere class, the bounding box class provides a much tighter fit for non-rotated rectangular objects.
- Because the bounding box class is axis aligned, you can make certain assumptions that result in collision checks between bounding boxes being quicker than a bounding box that can be rotated.

There are a few drawbacks of using the bounding box for collision detection.

- Rotating a bounding box causes it to no longer be axis aligned. Because of this, if you rotate a model being bounded, you will need to recreate the bounding box. Doing so can be slow, since all the points in an object are iterated through to get the bounding box. If the model has not changed orientation, you can translate the bounding box instead of recreating it.
- If the model being bounded is not aligned to the axis, the bounding box will have some empty space. The amount of empty space will be greatest when the object is rotated 45 degrees from an axis.
- Empty space in the bounding box can result in false positives when checking for collision.

See Also

Concepts

[Collision Detection Overview](#)

[Math Overview](#)

Tasks

[How To: Detect Whether Two Models Collide](#)

[How To: Position the Camera to View All Objects in a Scene](#)

Reference

[BoundingBox Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




BoundingBox Members

The following tables list the members exposed by the BoundingBox type.














Public Constructors

	Name	Description
	BoundingBox	Creates an instance of BoundingBox .



Public Fields

	Name	Description
	CornerCount	Specifies the total number of corners (8) in the BoundingBox .
	Max	The maximum point the BoundingBox contains.
	Min	The minimum point the BoundingBox contains.

Public Methods

	Name	Description
	Contains	Overloaded. Tests whether the BoundingBox overlaps another bounding volume.
	CreateFromPoints	Creates the smallest BoundingBox that will contain a group of points.
	CreateFromSphere	Overloaded. Creates the smallest BoundingBox that will contain the specified BoundingSphere .
	CreateMerged	Overloaded. Creates the smallest BoundingBox that contains the two specified BoundingBox instances.
	Equals	Overloaded. Determines whether two instances of BoundingBox are equal.
	GetCorners	Overloaded. Gets an array of points that make up the corners of the BoundingBox .
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the current BoundingBox intersects with another bounding volume.
	op_Equality	Determines whether two instances of BoundingBox are equal.
	op_Inequality	Determines whether two instances of BoundingBox are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current BoundingBox .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also




Reference

[BoundingBox Structure](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox Fields

Public Fields

	Name	Description
	CornerCount	Specifies the total number of corners (8) in the BoundingBox .
	Max	The maximum point the BoundingBox contains.
	Min	The minimum point the BoundingBox contains.

See Also

Reference

[BoundingBox Structure](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Corners Field

Specifies the total number of corners (8) in the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int Corners
```

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Max Field

The maximum point the [BoundingBox](#) contains.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Max
```

Remarks The maximum point is the point with components that are greater than or equal to the components of every other point within the [BoundingBox](#). Since a [BoundingBox](#) is axis aligned the maximum point is the upper-right corner of the front face of the [BoundingBox](#) when looking at it from the positive z direction.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Min Field

The minimum point the [BoundingBox](#) contains.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Min
```

Remarks The minimum point is the point with components that are less than or equal to the components of every other point within the [BoundingBox](#). Since a [BoundingBox](#) is axis aligned the minimum point is the lower-left corner of the back face of the [BoundingBox](#) when looking at it from the positive z direction.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox Constructor

Creates an instance of [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingBox (  
    Vector3 min,  
    Vector3 max  
)
```

Parameters

min

The minimum point the [BoundingBox](#) includes.

max

The maximum point the [BoundingBox](#) includes.

Remarks

The [BoundingBox Structure](#) represents the space occupied by a box. The bounding box class is axis aligned. Each face of the bounding box is perpendicular to the x-axis, the y-axis, or the z-axis.

There are several benefits of using the bounding box for collision detection.

- The bounding box class fits rectangular shapes aligned with the axis very well. Compared to the bounding sphere class, the bounding box class provides a much tighter fit for non-rotated rectangular objects.
- Because the bounding box class is axis aligned, you can make certain assumptions that result in collision checks between bounding boxes being quicker than a bounding box that can be rotated.

There are a few drawbacks of using the bounding box for collision detection.

- Rotating a bounding box causes it to no longer be axis aligned. Because of this, if you rotate a model being bounded, you will need to recreate the bounding box. Doing so can be slow, since all the points in an object are iterated through to get the bounding box. If the model has not changed orientation, you can translate the bounding box instead of recreating it.
- If the model being bounded is not aligned to the axis, the bounding box will have some empty space. The amount of empty space will be greatest when the object is rotated 45 degrees from an axis.
- Empty space in the bounding box can result in false positives when checking for collision.

See Also

Concepts

[Collision Detection Overview](#)

[Math Overview](#)

Tasks

[How To: Detect Whether Two Models Collide](#)

[How To: Position the Camera to View All Objects in a Scene](#)

Reference

[BoundingBox Structure](#)














[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox Methods

Public Methods

	Name	Description
	Contains	Overloaded. Tests whether the BoundingBox overlaps another bounding volume.
	CreateFromPoints	Creates the smallest BoundingBox that will contain a group of points.
	CreateFromSphere	Overloaded. Creates the smallest BoundingBox that will contain the specified BoundingBoxSphere .
	CreateMerged	Overloaded. Creates the smallest BoundingBox that contains the two specified BoundingBox instances.
	Equals	Overloaded. Determines whether two instances of BoundingBox are equal.
	GetCorners	Overloaded. Gets an array of points that make up the corners of the BoundingBox .
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the current BoundingBox intersects with another bounding volume.
	op_Equality	Determines whether two instances of BoundingBox are equal.
	op_Inequality	Determines whether two instances of BoundingBox are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current BoundingBox .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BoundingBox Structure](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Contains Method

Tests whether the [BoundingBox](#) overlaps another bounding volume.

Overload List

Name	Description
BoundingBox.Contains (BoundingBox)	Tests whether the BoundingBox contains another BoundingBox .
BoundingBox.Contains (BoundingBox, ContainmentType)	Tests whether the BoundingBox contains a BoundingBox .
BoundingBox.Contains (BoundingBoxFrustum)	Tests whether the BoundingBox contains a BoundingBoxFrustum .
BoundingBox.Contains (BoundingBoxSphere)	Tests whether the BoundingBox contains a BoundingBoxSphere .
BoundingBox.Contains (BoundingBoxSphere, ContainmentType)	Tests whether the BoundingBox contains a BoundingBoxSphere .
BoundingBox.Contains (Vector3)	Tests whether the BoundingBox contains a point.
BoundingBox.Contains (Vector3, ContainmentType)	Tests whether the BoundingBox contains a point.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Contains Method (BoundingBox)

Tests whether the [BoundingBox](#) contains another [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to test for overlap.

Return Value

Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox, ContainmentType)

Tests whether the [BoundingBox](#) contains a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref BoundingBox box,  
    out ContainmentType result  
)
```

Parameters

box

The [BoundingBox](#) to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingFrustum)

Tests whether the [BoundingBox](#) contains a [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingFrustum frustum  
)
```

Parameters

frustum

The [BoundingFrustum](#) to test for overlap.

Return Value

Enumeration indicating the extent of overlap.

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingSphere)

Tests whether the [BoundingBox](#) contains a [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to test for overlap.

Return Value

Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox, ContainmentType)

Tests whether the [BoundingBox](#) contains a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref BoundingBox sphere,  
    out ContainmentType result  
)
```

Parameters

sphere

The [BoundingBox](#) to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (Vector3)

Tests whether the [BoundingBox](#) contains a point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    Vector3 point  
)
```

Parameters

point

The point to test for overlap.

Return Value

Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (Vector3, ContainmentType)

Tests whether the [BoundingBox](#) contains a point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref Vector3 point,  
    out ContainmentType result  
)
```

Parameters

point

The point to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromPoints Method

Creates the smallest [BoundingBox](#) that will contain a group of points.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateFromPoints (  
    IEnumerable<Vector3> points  
)
```

Parameters

points

A list of points the [BoundingBox](#) should contain.

Return Value

The created [BoundingBox](#).

Exceptions

Exception type	Condition
ArgumentException	There are no points in <i>points</i> .
ArgumentNullException	<i>points</i> is null .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromSphere Method

Creates the smallest [BoundingBox](#) that will contain the specified [BoundingSphere](#).

Overload List

Name	Description
BoundingBox.CreateFromSphere (BoundingSphere)	Creates the smallest BoundingBox that will contain the specified BoundingSphere .
BoundingBox.CreateFromSphere (BoundingSphere, BoundingBox)	Creates the smallest BoundingBox that will contain the specified BoundingSphere .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.CreateFromSphere Method (BoundingSphere)

Creates the smallest [BoundingBox](#) that will contain the specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateFromSphere (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to contain.

Return Value

The created [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromSphere Method (BoundingSphere, BoundingBox)

Creates the smallest [BoundingBox](#) that will contain the specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromSphere (  
    ref BoundingSphere sphere,  
    out BoundingBox result  
)
```

Parameters

sphere

The [BoundingSphere](#) to contain.

result

[[OutAttribute](#)] The created [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateMerged Method

Creates the smallest [BoundingBox](#) that contains the two specified [BoundingBox](#) instances.

Overload List

Name	Description
BoundingBox.CreateMerged (BoundingBox, BoundingBox)	Creates the smallest BoundingBox that contains the two specified BoundingBox instances.
BoundingBox.CreateMerged (BoundingBox, BoundingBox, BoundingBox)	Creates the smallest BoundingBox that contains the two specified BoundingBox instances.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.CreateMerged Method (BoundingBox, BoundingBox)

Creates the smallest [BoundingBox](#) that contains the two specified [BoundingBox](#) instances.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateMerged (  
    BoundingBox original,  
    BoundingBox additional  
)
```

Parameters

original

One of the [BoundingBoxes](#) to contain.

additional

One of the [BoundingBoxes](#) to contain.

Return Value

The created [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateMerged Method (BoundingBox, BoundingBox, BoundingBox)

Creates the smallest [BoundingBox](#) that contains the two specified [BoundingBox](#) instances.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateMerged (  
    ref BoundingBox original,  
    ref BoundingBox additional,  
    out BoundingBox result  
)
```

Parameters

original

One of the [BoundingBox](#) instances to contain.

additional

One of the [BoundingBox](#) instances to contain.

result

[[OutAttribute](#)] The created [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Equals Method

Determines whether two instances of [BoundingBox](#) are equal.

Overload List

Name	Description
BoundingBox.Equals (BoundingBox)	Determines whether two instances of BoundingBox are equal.
BoundingBox.Equals (Object)	Determines whether two instances of BoundingBox are equal.
BoundingBox.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Equals Method (BoundingBox)

Determines whether two instances of [BoundingBox](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    BoundingBox other  
)
```

Parameters

other

The [BoundingBox](#) to compare with the current [BoundingBox](#).

Return Value

true if the specified [BoundingBox](#) is equal to the current [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Equals Method (Object)

Determines whether two instances of [BoundingBox](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [BoundingBox](#).

Return Value

true if the specified [Object](#) is equal to the current [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.GetCorners Method

Gets an array of points that make up the corners of the [BoundingBox](#).

Overload List

Name	Description
BoundingBox.GetCorners ()	Gets an array of points that make up the corners of the BoundingBox .
BoundingBox.GetCorners (Vector3[])	Gets the array of points that make up the corners of the BoundingBox .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.GetCorners Method ()

Gets an array of points that make up the corners of the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3[] GetCorners ()
```

Return Value

An array of [Vector3](#) points that represent the corners of the [BoundingBox](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	You have to have at least 8 elements to copy corners.

Remarks

The points returned correspond to the corners of the [BoundingBox](#) faces that are perpendicular to the z-axis. The near face is the face with the larger z value, and the far face is the face with the smaller z value. Points 0 to 3 correspond to the near face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction. Points 4 to 7 correspond to the far face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.GetCorners Method (Vector3[])

Gets the array of points that make up the corners of the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetCorners (
    Vector3[] corners
)
```

Parameters

corners

An existing array of at least 8 [Vector3](#) points where the corners of the [BoundingBox](#) are written.

Exceptions

Exception type	Condition
ArgumentNullException	<i>corners</i> is null .
ArgumentOutOfRangeException	You have to have at least 8 elements to copy corners.

Remarks

The points returned correspond to the corners of the [BoundingBox](#) faces that are perpendicular to the z-axis. The near face is the face with the larger z value, and the far face is the face with the smaller z value. Points 0 to 3 correspond to the near face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction. Points 4 to 7 correspond to the far face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method

Checks whether the current [BoundingBox](#) intersects with another bounding volume.

Overload List

Name	Description
BoundingBox.Intersects (BoundingBox)	Checks whether the current BoundingBox intersects another BoundingBox .
BoundingBox.Intersects (BoundingBox, Boolean)	Checks whether the current BoundingBox intersects another BoundingBox .
BoundingBox.Intersects (BoundingBoxFrustum)	Checks whether the current BoundingBox intersects a BoundingBoxFrustum .
BoundingBox.Intersects (BoundingBoxSphere)	Checks whether the current BoundingBox intersects a BoundingBoxSphere .
BoundingBox.Intersects (BoundingBoxSphere, Boolean)	Checks whether the current BoundingBox intersects a BoundingBoxSphere .
BoundingBox.Intersects (Plane)	Checks whether the current BoundingBox intersects a Plane .
BoundingBox.Intersects (Plane, PlaneIntersectionType)	Checks whether the current BoundingBox intersects a Plane .
BoundingBox.Intersects (Ray)	Checks whether the current BoundingBox intersects a Ray .
BoundingBox.Intersects (Ray, Nullable<Single>)	Checks whether the current BoundingBox intersects a Ray .

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Intersects Method (BoundingBox)

Checks whether the current [BoundingBox](#) intersects another [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with.

Return Value

true if the [BoundingBox](#)s intersect; **false** otherwise.

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox, Boolean)

Checks whether the current [BoundingBox](#) intersects another [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox box,  
    out bool result  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] **true** if the [BoundingBox](#) instances intersect; **false** otherwise.

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingFrustum)

Checks whether the current [BoundingBox](#) intersects a [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingFrustum frustum  
)
```

Parameters

frustum

The [BoundingFrustum](#) to check for intersection with.

Return Value

true if the [BoundingBox](#) and [BoundingFrustum](#) intersect; **false** otherwise.

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingSphere)

Checks whether the current [BoundingBox](#) intersects a [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to check for intersection with.

Return Value

true if the [BoundingBox](#) and [BoundingSphere](#) intersect; **false** otherwise.

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox, Boolean)

Checks whether the current [BoundingBox](#) intersects a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox sphere,  
    out bool result  
)
```

Parameters

sphere

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] **true** if the [BoundingBox](#) and [BoundingBox](#) intersect; **false** otherwise.

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Plane)

Checks whether the current [BoundingBox](#) intersects a [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneIntersectionType Intersects (  
    Plane plane  
)
```

Parameters

plane

The [Plane](#) to check for intersection with.

Return Value

An enumeration indicating whether the [BoundingBox](#) intersects the [Plane](#).

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Plane, PlaneIntersectionType)

Checks whether the current [BoundingBox](#) intersects a [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Plane plane,  
    out PlaneIntersectionType result  
)
```

Parameters

plane

The [Plane](#) to check for intersection with.

result

[[OutAttribute](#)] An enumeration indicating whether the [BoundingBox](#) intersects the [Plane](#).

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Ray)

Checks whether the current [BoundingBox](#) intersects a [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    Ray ray  
)
```

Parameters

ray

The [Ray](#) to check for intersection with.

Return Value

Distance at which the ray intersects the [BoundingBox](#), or **null** if there is no intersection.

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Ray, Nullable<Single>)

Checks whether the current [BoundingBox](#) intersects a [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Ray ray,  
    out Nullable<float> result  
)
```

Parameters

ray

The [Ray](#) to check for intersection with.

result

[[OutAttribute](#)] Distance at which the ray intersects the [BoundingBox](#), or **null** if there is no intersection.

See Also

Tasks

[How To: Detect Whether Two Models Collide](#)

Concepts

[Collision Detection Overview](#)

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.op_Equality Method

Determines whether two instances of [BoundingBox](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    BoundingBox a,  
    BoundingBox b  
)
```

Parameters

a

[BoundingBox](#) to compare.

b

[BoundingBox](#) to compare.

Return Value

true if the two [BoundingBox](#)s are equal; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.op_Inequality Method

Determines whether two instances of [BoundingBox](#) are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    BoundingBox a,  
    BoundingBox b  
)
```

Parameters

a

The object to the left of the inequality operator.

b

The object to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.ToString Method

Returns a [String](#) that represents the current [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of the current [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum Class

Defines a frustum and helps determine whether forms intersect with it.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

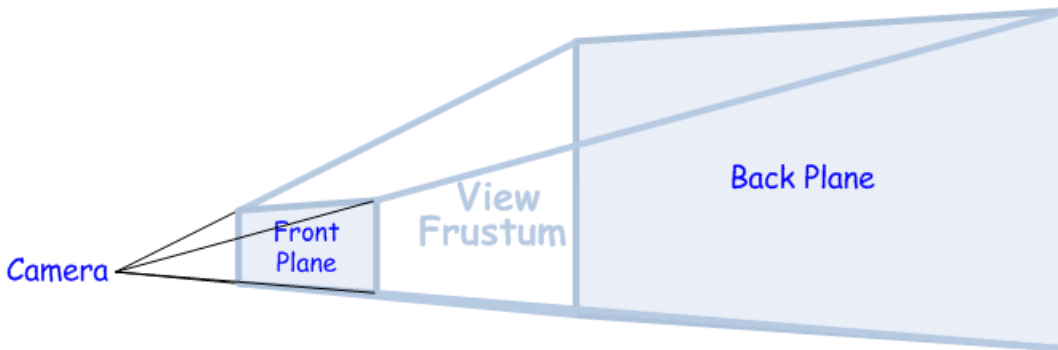
Syntax

C#

```
[TypeConverterAttribute(System.ComponentModel.ExpandableObjectConverter, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089)]
[SerializableAttribute]
public class BoundingFrustum : IEquatable<BoundingFrustum>
```

Remarks

A *frustum* in computer graphics is generally a volume of 3D space, defined as the part of a rectangular pyramid that lies between two planes perpendicular to its center line. A frustum is often used to represent what a "camera" sees in your 3D space—for example:



The **BoundingFrustum** class allows you to define a bounding frustum using a combined matrix that is generally the product of a view matrix and a projection matrix.

You can query a **BoundingFrustum** object for any one of its bounding planes, for its corners, and for whether it intersects with a given object. Since objects that don't intersect with your view frustum generally don't need to be rendered, culling them quickly can save you a lot of rendering time.

See Also

Reference

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


BoundingFrustum Members

The following tables list the members exposed by the BoundingFrustum type.








Public Constructors

Name	Description
 BoundingFrustum	Creates a new instance of BoundingFrustum .











Public Fields

Name	Description
 CornerCount	Specifies the total number of corners (8) in the BoundingFrustum .



Public Properties

Name	Description
 Bottom	Gets the bottom plane of the BoundingFrustum .
 Far	Gets the far plane of the BoundingFrustum .
 Left	Gets the left plane of the BoundingFrustum .
 Matrix	Gets or sets the Matrix that describes this bounding frustum.
 Near	Gets the near plane of the BoundingFrustum .
 Right	Gets the right plane of the BoundingFrustum .
 Top	Gets the top plane of the BoundingFrustum .

Public Methods

Name	Description
 Contains	Overloaded. Checks whether the current BoundingFrustum contains a specified bounding volume.
 Equals	Overloaded. Determines whether two instances of BoundingFrustum are equal.
 GetCorners	Overloaded. Gets an array of points that make up the corners of the BoundingFrustum .
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Intersects	Overloaded. Checks whether the current BoundingFrustum intersects a specified volume.
 Op_Equality	Determines whether two instances of BoundingFrustum are equal.
 Op_Inequality	Determines whether two instances of BoundingFrustum are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a String that represents the current BoundingFrustum .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also


Reference

[BoundingFrustum Class](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingFrustum Fields

Public Fields

	Name	Description
	CornerCount	Specifies the total number of corners (8) in the BoundingFrustum .

See Also

Reference

[BoundingFrustum Class](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingFrustum.CornerCount Field

Specifies the total number of corners (8) in the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int CornerCount
```

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum Constructor

Creates a new instance of [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingFrustum (  
    Matrix value  
)
```

Parameters

value

Combined matrix that usually takes view × projection matrix.

Remarks

You can use [BoundingFrustum Class](#) to create a bounding volume that corresponds to the space visible to the camera. You create a bounding frustum from the combined view and projection matrix that the camera is using currently. If the camera moves or rotates, you need to recreate the bounding frustum. The bounding frustum isn't used to determine when two objects collide, but rather when an object is in collision with the volume of space viewable by the camera. Objects that do not intersect and are not contained by the bounding frustum are not visible to the camera and don't need to be drawn. For complex models, this can save the graphics card a lot of work.

See Also

Concepts

[Collision Detection Overview](#)

Reference

[BoundingFrustum Class](#)













[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum Methods

Public Methods

	Name	Description
	Contains	Overloaded. Checks whether the current BoundingFrustum contains a specified bounding volume.
	Equals	Overloaded. Determines whether two instances of BoundingFrustum are equal.
	GetCorners	Overloaded. Gets an array of points that make up the corners of the BoundingFrustum .
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the current BoundingFrustum intersects a specified volume.
 	Op_Equality	Determines whether two instances of BoundingFrustum are equal.
 	Op_Inequality	Determines whether two instances of BoundingFrustum are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current BoundingFrustum .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BoundingFrustum Class](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Contains Method

Checks whether the current [BoundingBox](#) contains a specified bounding volume.

Overload List

Name	Description
BoundingBox.Contains (BoundingBox)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBox, ContainmentType)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBox)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBox)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBox, ContainmentType)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (Vector3)	Checks whether the current BoundingBox contains the specified point.
BoundingBox.Contains (Vector3, ContainmentType)	Checks whether the current BoundingBox contains the specified point.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Contains Method (BoundingBox)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to check against the current [BoundingBox](#).

Return Value

Enumeration indicating the relationship of the current [BoundingBox](#) to the specified [BoundingBox](#).

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox, ContainmentType)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref BoundingBox box,  
    out ContainmentType result  
)
```

Parameters

box

The [BoundingBox](#) to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingBox frustum  
)
```

Parameters

frustum

The [BoundingBox](#) to check against the current [BoundingBox](#).

Return Value

Enumeration indicating the relationship of the current [BoundingBox](#) to the specified [BoundingBox](#).

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingSphere)

Checks whether the current [BoundingBox](#) contains the specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to check against the current [BoundingBox](#).

Return Value

Enumeration indicating the relationship of the current [BoundingBox](#) to the specified [BoundingSphere](#).

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.Contains Method (BoundingSphere, ContainmentType)

Checks whether the current [BoundingFrustum](#) contains the specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref BoundingSphere sphere,  
    out ContainmentType result  
)
```

Parameters

sphere

The [BoundingSphere](#) to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (Vector3)

Checks whether the current [BoundingBox](#) contains the specified point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    Vector3 point  
)
```

Parameters

point

The point to check against the current [BoundingBox](#).

Return Value

Enumeration indicating the relationship of the current [BoundingBox](#) to the specified point.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (Vector3, ContainmentType)

Checks whether the current [BoundingBox](#) contains the specified point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref Vector3 point,  
    out ContainmentType result  
)
```

Parameters

point

The point to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Equals Method

Determines whether two instances of [BoundingBox](#) are equal.

Overload List

Name	Description
BoundingBox.Equals (BoundingBox)	Determines whether the specified BoundingBox is equal to the current BoundingBox .
BoundingBox.Equals (Object)	Determines whether the specified Object is equal to the BoundingBox .
BoundingBox.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Equals Method (BoundingBox)

Determines whether the specified [BoundingBox](#) is equal to the current [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    BoundingBox other  
)
```

Parameters

other

The [BoundingBox](#) to compare with the current [BoundingBox](#).

Return Value

true if the specified [BoundingBox](#) is equal to the current [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Equals Method (Object)

Determines whether the specified [Object](#) is equal to the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [BoundingBox](#).

Return Value

true if the specified [Object](#) is equal to the current [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.GetCorners Method

Gets an array of points that make up the corners of the [BoundingBox](#).

Overload List

Name	Description
BoundingBox.GetCorners ()	Gets an array of points that make up the corners of the BoundingBox .
BoundingBox.GetCorners (Vector3[])	Gets an array of points that make up the corners of the BoundingBox .

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.GetCorners Method ()

Gets an array of points that make up the corners of the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3[] GetCorners ()
```

Return Value

Array of [Vector3](#) points that make up the corners of the [BoundingBox](#).

Remarks

The points returned correspond to the corners of the [BoundingBox](#) faces that are perpendicular to the z-axis. The near face is the face with the larger z value, and the far face is the face with the smaller z value. Points 0 to 3 correspond to the near face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction. Points 4 to 7 correspond to the far face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.GetCorners Method (Vector3[])

Gets an array of points that make up the corners of the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetCorners (
    Vector3[] corners
)
```

Parameters

corners

An existing array of at least 8 [Vector3](#) points where the corners of the [BoundingFrustum](#) are written.

Exceptions

Exception type	Condition
ArgumentNullException	<i>corners</i> is null .
ArgumentOutOfRangeException	You have to have at least 8 elements to copy corners.

Remarks

The points returned correspond to the corners of the [BoundingFrustum](#) faces that are perpendicular to the z-axis. The near face is the face with the larger z value, and the far face is the face with the smaller z value. Points 0 to 3 correspond to the near face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction. Points 4 to 7 correspond to the far face in a clockwise order starting at its upper-left corner when looking toward the origin from the positive z direction.

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [BoundingBox](#).

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method

Checks whether the current [BoundingBox](#) intersects a specified volume.

Overload List

Name	Description
BoundingBox.Intersects (BoundingBox)	Checks whether the current BoundingBox intersects the specified BoundingBox .
BoundingBox.Intersects (BoundingBox, Boolean)	Checks whether the current BoundingBox intersects a BoundingBox .
BoundingBox.Intersects (BoundingBox)	Checks whether the current BoundingBox intersects the specified BoundingBox .
BoundingBox.Intersects (BoundingBox)	Checks whether the current BoundingBox intersects the specified BoundingBox .
BoundingBox.Intersects (BoundingBox, Boolean)	Checks whether the current BoundingBox intersects a BoundingBox .
BoundingBox.Intersects (Plane)	Checks whether the current BoundingBox intersects the specified Plane .
BoundingBox.Intersects (Plane, PlaneIntersectionType)	Checks whether the current BoundingBox intersects a Plane .
BoundingBox.Intersects (Ray)	Checks whether the current BoundingBox intersects the specified Ray .
BoundingBox.Intersects (Ray, Nullable<Single>)	Checks whether the current BoundingBox intersects a Ray .

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Intersects Method (BoundingBox)

Checks whether the current [BoundingBox](#) intersects the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection.

Return Value

true if the [BoundingBox](#) intersects the [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox, Boolean)

Checks whether the current [BoundingBox](#) intersects a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox box,  
    out bool result  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] **true** if the [BoundingBox](#) and [BoundingBox](#) intersect; **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox)

Checks whether the current [BoundingBox](#) intersects the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingBox frustum  
)
```

Parameters

frustum

The [BoundingBox](#) to check for intersection.

Return Value

true if the current [BoundingBox](#) intersects the specified [BoundingBox](#); **false** otherwise.

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingSphere)

Checks whether the current [BoundingBox](#) intersects the specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to check for intersection.

Return Value

true if the [BoundingBox](#) intersects the [BoundingSphere](#); **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox, Boolean)

Checks whether the current [BoundingBox](#) intersects a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox sphere,  
    out bool result  
)
```

Parameters

sphere

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] **true** if the [BoundingBox](#) and [BoundingBox](#) intersect; **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Plane)

Checks whether the current [BoundingBox](#) intersects the specified [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneIntersectionType Intersects (  
    Plane plane  
)
```

Parameters

plane

The [Plane](#) to check for intersection.

Return Value

An enumeration indicating whether [BoundingBox](#) intersects the specified [Plane](#).

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Plane, PlaneIntersectionType)

Checks whether the current [BoundingBox](#) intersects a [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Plane plane,  
    out PlaneIntersectionType result  
)
```

Parameters

plane

The [Plane](#) to check for intersection with.

result

[[OutAttribute](#)] An enumeration indicating whether the [BoundingBox](#) intersects the [Plane](#).

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Ray)

Checks whether the current [BoundingBox](#) intersects the specified [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    Ray ray  
)
```

Parameters

ray

The [Ray](#) to check for intersection.

Return Value

Distance at which the ray intersects the [BoundingBox](#) or **null** if there is no intersection.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Ray, Nullable<Single>)

Checks whether the current [BoundingBox](#) intersects a [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Ray ray,  
    out Nullable<float> result  
)
```

Parameters

ray

The [Ray](#) to check for intersection with.

result

[[OutAttribute](#)] Distance at which the ray intersects the [BoundingBox](#) or **null** if there is no intersection.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.op_Equality Method

Determines whether two instances of [BoundingBox](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    BoundingBox a,  
    BoundingBox b  
)
```

Parameters

a

The [BoundingBox](#) to the left of the equality operator.

b

The [BoundingBox](#) to the right of the equality operator.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.op_Inequality Method

Determines whether two instances of [BoundingBox](#) are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    BoundingBox a,  
    BoundingBox b  
)
```

Parameters

a

The [BoundingBox](#) to the left of the inequality operator.

b

The [BoundingBox](#) to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.ToString Method

Returns a [String](#) that represents the current [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of the current [BoundingBox](#).

See Also

Reference

[BoundingBox Class](#)








[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum Properties

Public Properties

	Name	Description
	Bottom	Gets the bottom plane of the BoundingFrustum .
	Far	Gets the far plane of the BoundingFrustum .
	Left	Gets the left plane of the BoundingFrustum .
	Matrix	Gets or sets the Matrix that describes this bounding frustum.
	Near	Gets the near plane of the BoundingFrustum .
	Right	Gets the right plane of the BoundingFrustum .
	Top	Gets the top plane of the BoundingFrustum .

See Also

Reference

[BoundingFrustum Class](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingFrustum.Bottom Property

Gets the bottom plane of the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Bottom { get; }
```

Property Value

Returns the bottom plane of the [BoundingFrustum](#).

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.Far Property

Gets the far plane of the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Far { get; }
```

Property Value

Returns the far plane of the [BoundingFrustum](#).

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.Left Property

Gets the left plane of the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Left { get; }
```

Property Value

Returns the left plane of the [BoundingFrustum](#).

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.Matrix Property

Gets or sets the [Matrix](#) that describes this bounding frustum.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix Matrix { get; set; }
```

Property Value

The [Matrix](#) that describes this bounding frustum.

Remarks

This property can be used to reset an existing [BoundingFrustum](#) to new values, instead of creating a new one.

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.Near Property

Gets the near plane of the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Near { get; }
```

Property Value

Returns the near plane of the [BoundingFrustum](#).

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingFrustum.Right Property

Gets the right plane of the [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Right { get; }
```

Property Value

Returns the right plane of the [BoundingFrustum](#).

See Also

Reference

[BoundingFrustum Class](#)

[BoundingFrustum Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Top Property

Gets the top plane of the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Top { get; }
```

Property Value

Returns the top plane of the [BoundingBox](#).

See Also

Reference

[BoundingBox Class](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox Structure

Defines a sphere.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.BoundingBoxConverter)")]
[SerializableAttribute]
public struct BoundingBox : IEquatable<BoundingBox>
```

See Also

Reference

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)


[How To: Position the Camera to View All Objects in a Scene](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



BoundingBox Members

The following tables list the members exposed by the BoundingBox type.





















Public Constructors

	Name	Description
	BoundingBox	Creates a new instance of BoundingBox .



Public Fields

	Name	Description
	Center	The center point of the sphere.
	Radius	The radius of the sphere.

Public Methods

	Name	Description
	Contains	Overloaded. Checks whether the current BoundingBox contains a specified bounding volume.
 	CreateFromBoundingBox	Overloaded. Creates the smallest BoundingBox that can contain a specified BoundingBox .
 	CreateFromFrustum	Creates the smallest BoundingBox that can contain a specified BoundingFrustum .
 	CreateFromPoints	Creates a BoundingBox that can contain a specified list of points.
 	CreateMerged	Overloaded. Creates a BoundingBox that contains the two specified BoundingBox instances.
	Equals	Overloaded. Determines whether two instances of BoundingBox are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the current BoundingBox intersects another bounding volume.
 	op_Equality	Determines whether two instances of BoundingBox are equal.
 	op_Inequality	Determines whether two instances of BoundingBox are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current BoundingBox .
	Transform	Overloaded. Translates and scales the BoundingBox using a given Matrix .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[BoundingBox Structure](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox Fields

Public Fields

	Name	Description
	Center	The center point of the sphere.
	Radius	The radius of the sphere.

See Also

Reference

[BoundingBox Structure](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Center Field

The center point of the sphere.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Center
```

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Radius Field

The radius of the sphere.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Radius
```

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox Constructor

Creates a new instance of [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingBox (
    Vector3 center,
    float radius
)
```

Parameters

center

Center point of the sphere.

radius

Radius of the sphere.

Exceptions

Exception type	Condition
ArgumentException	<i>radius</i> is less than zero. <i>radius</i> must be greater than or equal to zero.

See Also

Reference

[BoundingBox Structure](#)





















[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox Methods

Public Methods

	Name	Description
	Contains	Overloaded. Checks whether the current BoundingBox contains a specified bounding volume.
 	CreateFromBoundingBox	Overloaded. Creates the smallest BoundingBox that can contain a specified BoundingBox .
 	CreateFromFrustum	Creates the smallest BoundingBox that can contain a specified BoundingFrustum .
 	CreateFromPoints	Creates a BoundingBox that can contain a specified list of points.
 	CreateMerged	Overloaded. Creates a BoundingBox that contains the two specified BoundingBox instances.
	Equals	Overloaded. Determines whether two instances of BoundingBox are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the current BoundingBox intersects another bounding volume.
 	op_Equality	Determines whether two instances of BoundingBox are equal.
 	op_Inequality	Determines whether two instances of BoundingBox are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current BoundingBox .
	Transform	Overloaded. Translates and scales the BoundingBox using a given Matrix .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BoundingBox Structure](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Contains Method

Checks whether the current [BoundingBox](#) contains a specified bounding volume.

Overload List

Name	Description
BoundingBox.Contains (BoundingBox)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBox, ContainmentType)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBoxFrustum)	Checks whether the current BoundingBox contains the specified BoundingBoxFrustum .
BoundingBox.Contains (BoundingBox)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (BoundingBox, ContainmentType)	Checks whether the current BoundingBox contains the specified BoundingBox .
BoundingBox.Contains (Vector3)	Checks whether the current BoundingBox contains the specified point.
BoundingBox.Contains (Vector3, ContainmentType)	Checks whether the current BoundingBox contains the specified point.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Contains Method (BoundingBox)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to check against the current [BoundingBox](#).

Return Value

An enumeration indicating the relationship of the specified [BoundingBox](#) to the current [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox, ContainmentType)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref BoundingBox box,  
    out ContainmentType result  
)
```

Parameters

box

The [BoundingBox](#) to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingBox frustum  
)
```

Parameters

frustum

The [BoundingBox](#) to check against the current [BoundingBox](#).

Return Value

An enumeration indicating the relationship of the specified [BoundingBox](#) to the current [BoundingBox](#).

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    BoundingBox sphere  
)
```

Parameters

sphere

The [BoundingBox](#) to check against the current [BoundingBox](#).

Return Value

An enumeration indicating the relationship of the [BoundingBoxes](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (BoundingBox, ContainmentType)

Checks whether the current [BoundingBox](#) contains the specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref BoundingBox sphere,  
    out ContainmentType result  
)
```

Parameters

sphere

The [BoundingBox](#) to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (Vector3)

Checks whether the current [BoundingBox](#) contains the specified point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContainmentType Contains (  
    Vector3 point  
)
```

Parameters

point

The point to check against the current [BoundingBox](#).

Return Value

An enumeration indicating the relationship of the specified point to the current [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Contains Method (Vector3, ContainmentType)

Checks whether the current [BoundingBox](#) contains the specified point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref Vector3 point,  
    out ContainmentType result  
)
```

Parameters

point

The point to test for overlap.

result

[[OutAttribute](#)] Enumeration indicating the extent of overlap.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromBoundingBox Method

Creates the smallest [BoundingBox](#) that can contain a specified [BoundingBox](#).

Overload List

Name	Description
BoundingBox.CreateFromBoundingBox (BoundingBox)	Creates the smallest BoundingBox that can contain a specified BoundingBox .
BoundingBox.CreateFromBoundingBox (BoundingBox, BoundingBox)	Creates the smallest BoundingBox that can contain a specified BoundingBox .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.CreateFromBoundingBox Method (BoundingBox)

Creates the smallest [BoundingBox](#) that can contain a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateFromBoundingBox (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to create the [BoundingBox](#) from.

Return Value

The created [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Position the Camera to View All Objects in a Scene](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromBoundingBox Method (BoundingBox, BoundingBox)

Creates the smallest [BoundingBox](#) that can contain a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromBoundingBox (  
    ref BoundingBox box,  
    out BoundingBox result  
)
```

Parameters

box

The [BoundingBox](#) to create the [BoundingBox](#) from.

result

[[OutAttribute](#)] The created [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromFrustum Method

Creates the smallest [BoundingBox](#) that can contain a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateFromFrustum (  
    BoundingBox frustum  
)
```

Parameters

frustum

The [BoundingBox](#) to create the [BoundingBox](#) with.

Return Value

The created [BoundingBox](#).

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateFromPoints Method

Creates a [BoundingBox](#) that can contain a specified list of points.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateFromPoints (  
    IEnumerable<Vector3> points  
)
```

Parameters

points

List of points the [BoundingBox](#) must contain.

Return Value

The created [BoundingBox](#).

Exceptions

Exception type	Condition
ArgumentException	There are no points in <i>points</i> .
ArgumentNullException	<i>points</i> is null .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Tasks

[How To: Position the Camera to View All Objects in a Scene](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateMerged Method

Creates a [BoundingBox](#) that contains the two specified [BoundingBox](#) instances.

Overload List

Name	Description
BoundingBox.CreateMerged (BoundingBox, BoundingBox)	Creates a BoundingBox that contains the two specified BoundingBox instances.
BoundingBox.CreateMerged (BoundingBox, BoundingBox, BoundingBox)	Creates a BoundingBox that contains the two specified BoundingBox instances.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.CreateMerged Method (BoundingBox, BoundingBox)

Creates a [BoundingBox](#) that contains the two specified [BoundingBox](#) instances.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BoundingBox CreateMerged (  
    BoundingBox original,  
    BoundingBox additional  
)
```

Parameters

original

[BoundingBox](#) to be merged.

additional

[BoundingBox](#) to be merged.

Return Value

The created [BoundingBox](#).

Remarks

This method is used in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: "Ships" Passing in the Night](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Position the Camera to View All Objects in a Scene](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.CreateMerged Method (BoundingBox, BoundingBox, BoundingBox)

Creates a [BoundingBox](#) that contains the two specified [BoundingBox](#) instances.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateMerged (  
    ref BoundingBox original,  
    ref BoundingBox additional,  
    out BoundingBox result  
)
```

Parameters

original

[BoundingBox](#) to be merged.

additional

[BoundingBox](#) to be merged.

result

[[OutAttribute](#)] The created [BoundingBox](#).

Remarks

This method is used in the FuelCell game, a game developed by following a series of focused articles that discuss basic 3D game development. For more information, see [FuelCell: "Ships" Passing in the Night](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Equals Method

Determines whether two instances of [BoundingBox](#) are equal.

Overload List

Name	Description
BoundingBox.Equals (BoundingBox)	Determines whether the specified BoundingBox is equal to the current BoundingBox .
BoundingBox.Equals (Object)	Determines whether the specified Object is equal to the BoundingBox .
BoundingBox.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Equals Method (BoundingBox)

Determines whether the specified [BoundingBox](#) is equal to the current [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    BoundingBox other  
)
```

Parameters

other

The [BoundingBox](#) to compare with the current [BoundingBox](#).

Return Value

true if the specified [BoundingBox](#) is equal to the current [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Equals Method (Object)

Determines whether the specified [Object](#) is equal to the [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [BoundingBox](#).

Return Value

true if the specified [Object](#) is equal to the current [BoundingBox](#); **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method

Checks whether the current [BoundingBox](#) intersects another bounding volume.

Overload List

Name	Description
BoundingBox.Intersects (BoundingBox)	Checks whether the current BoundingBox intersects with a specified BoundingBox .
BoundingBox.Intersects (BoundingBox, Boolean)	Checks whether the current BoundingBox intersects a BoundingBox .
BoundingBox.Intersects (BoundingBoxFrustum)	Checks whether the current BoundingBox intersects with a specified BoundingBoxFrustum .
BoundingBox.Intersects (BoundingBox)	Checks whether the current BoundingBox intersects with a specified BoundingBox .
BoundingBox.Intersects (BoundingBox, Boolean)	Checks whether the current BoundingBox intersects another BoundingBox .
BoundingBox.Intersects (Plane)	Checks whether the current BoundingBox intersects with a specified Plane .
BoundingBox.Intersects (Plane, PlaneIntersectionType)	Checks whether the current BoundingBox intersects a Plane .
BoundingBox.Intersects (Ray)	Checks whether the current BoundingBox intersects with a specified Ray .
BoundingBox.Intersects (Ray, Nullable<Single>)	Checks whether the current BoundingBox intersects a Ray .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Intersects Method (BoundingBox)

Checks whether the current [BoundingBox](#) intersects with a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with the current [BoundingBox](#).

Return Value

true if the [BoundingBox](#) and [BoundingBox](#) intersect; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox, Boolean)

Checks whether the current [BoundingBox](#) intersects a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox box,  
    out bool result  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] **true** if the [BoundingBox](#) and [BoundingBox](#) intersect; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox)

Checks whether the current [BoundingBox](#) intersects with a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingBox frustum  
)
```

Parameters

frustum

The [BoundingBox](#) to check for intersection with the current [BoundingBox](#).

Return Value

true if the [BoundingBox](#) and [BoundingBox](#) intersect; **false** otherwise.

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox)

Checks whether the current [BoundingBox](#) intersects with a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    BoundingBox sphere  
)
```

Parameters

sphere

The [BoundingBox](#) to check for intersection with the current [BoundingBox](#).

Return Value

true if the [BoundingBoxes](#) intersect; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (BoundingBox, Boolean)

Checks whether the current [BoundingBox](#) intersects another [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox sphere,  
    out bool result  
)
```

Parameters

sphere

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] **true** if the [BoundingBox](#) instances intersect; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Plane)

Checks whether the current [BoundingBox](#) intersects with a specified [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneIntersectionType Intersects (  
    Plane plane  
)
```

Parameters

plane

The [Plane](#) to check for intersection with the current [BoundingBox](#).

Return Value

An enumeration indicating the relationship between the [BoundingBox](#) and the [Plane](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Plane, PlaneIntersectionType)

Checks whether the current [BoundingBox](#) intersects a [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Plane plane,  
    out PlaneIntersectionType result  
)
```

Parameters

plane

The [Plane](#) to check for intersection with.

result

[[OutAttribute](#)] An enumeration indicating whether the [BoundingBox](#) intersects the [Plane](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Ray)

Checks whether the current [BoundingBox](#) intersects with a specified [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    Ray ray  
)
```

Parameters

ray

The [Ray](#) to check for intersection with the current [BoundingBox](#).

Return Value

Distance at which the ray intersects the [BoundingBox](#) or **null** if there is no intersection.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Intersects Method (Ray, Nullable<Single>)

Checks whether the current [BoundingBox](#) intersects a [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Ray ray,  
    out Nullable<float> result  
)
```

Parameters

ray

The [Ray](#) to check for intersection with.

result

[[OutAttribute](#)] Distance at which the ray intersects the [BoundingBox](#) or **null** if there is no intersection.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.op_Equality Method

Determines whether two instances of [BoundingBox](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    BoundingBox a,  
    BoundingBox b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.op_Inequality Method

Determines whether two instances of [BoundingBox](#) are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    BoundingBox a,  
    BoundingBox b  
)
```

Parameters

a

The [BoundingBox](#) to the left of the inequality operator.

b

The [BoundingBox](#) to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.ToString Method

Returns a [String](#) that represents the current [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of the current [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Transform Method

Translates and scales the [BoundingBox](#) using a given [Matrix](#).

Overload List

Name	Description
BoundingBox.Transform (Matrix)	Translates and scales the BoundingBox using a given Matrix .
BoundingBox.Transform (Matrix, BoundingBox)	Translates and scales the BoundingBox using a given Matrix .

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[Microsoft.Xna.Framework Namespace](#)

BoundingBox.Transform Method (Matrix)

Translates and scales the [BoundingBox](#) using a given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingBox Transform (  
    Matrix matrix  
)
```

Parameters

matrix

A transformation matrix that might include translation, rotation, or uniform scaling. Note that [BoundingBox.Transform](#) will not return correct results if there are non-uniform scaling, shears, or other unusual transforms in this transformation matrix. This is because there is no way to shear or non-uniformly scale a sphere. Such an operation would cause the sphere to lose its shape as a sphere.

Return Value

The transformed [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[CreateScale](#)

[CreateTranslation](#)

[CreateWorld](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

BoundingBox.Transform Method (Matrix, BoundingBox)

Translates and scales the [BoundingBox](#) using a given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Transform (  
    ref Matrix matrix,  
    out BoundingBox result  
)
```

Parameters

matrix

A transformation matrix that might include translation, rotation, or uniform scaling. Note that [BoundingBox.Transform](#) will not return correct results if there are non-uniform scaling, shears, or other unusual transforms in this transformation matrix. This is because there is no way to shear or non-uniformly scale a sphere. Such an operation would cause the sphere to lose its shape as a sphere.

result

[[OutAttribute](#)] The transformed [BoundingBox](#).

See Also

Reference

[BoundingBox Structure](#)

[BoundingBox Members](#)

[CreateScale](#)

[CreateTranslation](#)

[CreateWorld](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContainmentType Enumeration

Indicates the extent to which bounding volumes intersect or contain one another.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ContainmentType
```

Members

Member name	Description
Contains	Indicates that one bounding volume completely contains the other.
Disjoint	Indicates there is no overlap between the bounding volumes.
Intersects	Indicates that the bounding volumes partially overlap.

See Also

Reference

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve Class

Stores an arbitrary collection of 2D [CurveKey](#) points, and provides methods for evaluating features of the curve they define.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute(System.ComponentModel.ExpandableObjectConverter, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089)]
[SerializableAttribute]
public class Curve
```

Remarks

The [Curve](#) class describes how two [float](#) values vary with respect to each other even when you cannot easily define their relationship by a formula. Instead, the relationship is defined by an ordered set of value pairs that exemplify it.

You specify a [Curve](#) by adding a sequence of [CurveKey](#) objects to its [Keys](#) collection, each of which must contain at minimum a [Position](#) greater than that of the preceding [CurveKey](#), and a corresponding [Value](#). In practice, the [Position](#) is almost always used to represent a point in time, while the [Value](#) property generally represents a coordinate value corresponding to that point in time.

To represent a time path in two or three dimensions, you can define two or three [Curve](#) objects, each of which corresponds to a different spatial axis. Coordinates of a given time's point on the resulting 2D or 3D curve are calculated very efficiently.

The [Curve.Evaluate](#) method returns a (coordinate) value corresponding to a (time) position that you specify. The position doesn't have to be contained in any of the [CurveKey](#) points, because [Curve](#) interpolates between points depending on the [CurveKey](#) characteristics you have defined.

Not only is [Curve](#) useful for controlling animation and spatial motion, it can also be used anywhere you want to define a response that changes over time. For example, if you are using a button to accelerate a car in your game, you might want to define a complex velocity curve that varies with how long the button is held down to simulate gear shifts.

A [CurveKey](#) point need not always be part of a smoothly varying curve; you can also set a [CurveKey](#) to represent a step or sharp inflection point.

See Also

Reference

[CurveKey](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)


[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





Curve Members

The following tables list the members exposed by the Curve type.










Public Constructors

	Name	Description
	Curve	Initializes a new instance of Curve .



Public Properties

	Name	Description
	IsConstant	Gets a value indicating whether the curve is constant.
	Keys	The points that make up the curve.
	PostLoop	Specifies how to handle weighting values that are greater than the last control point in the curve.
	PreLoop	Specifies how to handle weighting values that are less than the first control point in the curve.

Public Methods

	Name	Description
	Clone	Creates a copy of the Curve .
	ComputeTangent	Overloaded. Computes the tangents for a specified CurveKey in this Curve .
	ComputeTangents	Overloaded. Computes all tangents for all CurveKeys in the Curve .
	Equals	(Inherited from Object .)
	Evaluate	Finds the value at a position on the Curve .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Curve Class](#)

[Microsoft.Xna.Framework Namespace](#)

Curve Constructor

Initializes a new instance of [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Curve ()
```

See Also

Reference

[Curve Class](#)










[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve Methods

Public Methods

	Name	Description
	Clone	Creates a copy of the Curve .
	ComputeTangent	Overloaded. Computes the tangents for a specified CurveKey in this Curve .
	ComputeTangents	Overloaded. Computes all tangents for all CurveKeys in the Curve .
	Equals	(Inherited from Object .)
	Evaluate	Finds the value at a position on the Curve .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Curve Class](#)

[Microsoft.Xna.Framework Namespace](#)

Curve.Clone Method

Creates a copy of the [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Curve Clone ()
```

Return Value

The copy of the [Curve](#).

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.ComputeTangent Method

Computes the tangents for a specified [CurveKey](#) in this [Curve](#).

Overload List

Name	Description
Curve.ComputeTangent (Int32, CurveTangent)	Computes both the TangentIn and the TangentOut for a CurveKey specified by its index.
Curve.ComputeTangent (Int32, CurveTangent, CurveTangent)	Computes a specified type of TangentIn and a specified type of TangentOut for a given CurveKey .

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Curve.ComputeTangent Method (Int32, CurveTangent)

Computes both the [TangentIn](#) and the [TangentOut](#) for a [CurveKey](#) specified by its index.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ComputeTangent (  
    int keyIndex,  
    CurveTangent tangentType  
)
```

Parameters

keyIndex

The index of the [CurveKey](#) for which to compute tangents (in the [Keys](#) collection of the [Curve](#)).

tangentType

The type of tangents to compute (one of the types specified in the [CurveTangent](#) enumeration).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>keyIndex</i> is not a valid index.

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.ComputeTangent Method (Int32, CurveTangent, CurveTangent)

Computes a specified type of [TangentIn](#) and a specified type of [TangentOut](#) for a given [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ComputeTangent (  
    int keyIndex,  
    CurveTangent tangentInType,  
    CurveTangent tangentOutType  
)
```

Parameters

keyIndex

The index of the [CurveKey](#) for which to compute tangents (in the [Keys](#) collection of the [Curve](#)).

tangentInType

The type of [TangentIn](#) to compute (one of the types specified in the [CurveTangent](#) enumeration).

tangentOutType

The type of [TangentOut](#) to compute (one of the types specified in the [CurveTangent](#) enumeration).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>keyIndex</i> is not a valid index.

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.ComputeTangents Method

Computes all tangents for all [CurveKeys](#) in the [Curve](#).

Overload List

Name	Description
Curve.ComputeTangents (CurveTangent)	Computes all tangents for all CurveKeys in this Curve , using a specified tangent type for both TangentIn and TangentOut .
Curve.ComputeTangents (CurveTangent, CurveTangent)	Computes all tangents for all CurveKeys in this Curve , using different tangent types for TangentOut and TangentIn .

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Curve.ComputeTangents Method (CurveTangent)

Computes all tangents for all [CurveKeys](#) in this [Curve](#), using a specified tangent type for both [TangentIn](#) and [TangentOut](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ComputeTangents (  
    CurveTangent tangentType  
)
```

Parameters

tangentType

The type of [TangentOut](#) and [TangentIn](#) to compute (one of the types specified in the [CurveTangent](#) enumeration).

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.ComputeTangents Method (CurveTangent, CurveTangent)

Computes all tangents for all [CurveKeys](#) in this [Curve](#), using different tangent types for [TangentOut](#) and [TangentIn](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ComputeTangents (  
    CurveTangent tangentInType,  
    CurveTangent tangentOutType  
)
```

Parameters

tangentInType

The type of [TangentIn](#) to compute (one of the types specified in the [CurveTangent](#) enumeration).

tangentOutType

The type of [TangentOut](#) to compute (one of the types specified in the [CurveTangent](#) enumeration).

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.Evaluate Method

Finds the value at a position on the [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Evaluate (  
    float position  
)
```

Parameters

position

The position on the [Curve](#).

Return Value

Value at the position on the [Curve](#).

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)





Programming Guide

[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve Properties

Public Properties

	Name	Description
	IsConstant	Gets a value indicating whether the curve is constant.
	Keys	The points that make up the curve.
	PostLoop	Specifies how to handle weighting values that are greater than the last control point in the curve.
	PreLoop	Specifies how to handle weighting values that are less than the first control point in the curve.

See Also

Reference

[Curve Class](#)

[Microsoft.Xna.Framework Namespace](#)

Curve.IsConstant Property

Gets a value indicating whether the curve is constant.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConstant { get; }
```

Property Value

true is the curve is constant (has one or fewer points); **false** otherwise.

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.Keys Property

The points that make up the curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKeyCollection Keys { get; }
```

Property Value

Points that make up the curve.

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.PostLoop Property

Specifies how to handle weighting values that are greater than the last control point in the curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveLoopType PostLoop { get; set; }
```

Property Value

Specifies how to handle weighting values.

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Curve.PreLoop Property

Specifies how to handle weighting values that are less than the first control point in the curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveLoopType PreLoop { get; set; }
```

Property Value

Specifies how to handle weighting values.

See Also

Reference

[Curve Class](#)

[Curve Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveContinuity Enumeration

Defines the continuity of [CurveKeys](#) on a [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CurveContinuity
```

Members

Member name	Description
Smooth	Interpolation can be used between this CurveKey and the next.
Step	Interpolation cannot be used between this CurveKey and the next. Specifying a position between the two points returns this point.

See Also

Reference

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey Class

Represents a point in a multi-point curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute(System.ComponentModel.ExpandableObjectConverter, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089)]
[SerializableAttribute]
public class CurveKey : IEquatable<CurveKey>, IComparable<CurveKey>
```

See Also

Reference

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)


[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






CurveKey Members

The following tables list the members exposed by the CurveKey type.










Public Constructors

	Name	Description
	CurveKey	Overloaded. Initializes a new instance of CurveKey .



Public Properties

	Name	Description
	Continuity	Describes whether the segment between this point and the next point in the curve is discrete or continuous.
	Position	Position of the CurveKey in the curve.
	TangentIn	Describes the tangent when approaching this point from the previous point in the curve.
	TangentOut	Describes the tangent when leaving this point to the next point in the curve.
	Value	Describes the value of this point.

Public Methods

	Name	Description
	Clone	Creates a copy of the CurveKey .
	CompareTo	Compares this instance to another CurveKey and returns an indication of their relative values.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Returns the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two CurveKey instances are equal.
	Op_Inequality	Determines whether two CurveKey instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CurveKey Class](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKey Constructor

Initializes a new instance of [CurveKey](#).

Overload List

Name	Description
CurveKey (Single, Single)	Initializes a new instance of CurveKey .
CurveKey (Single, Single, Single, Single)	Initializes a new instance of CurveKey .
CurveKey (Single, Single, Single, Single, CurveContinuity)	Initializes a new instance of CurveKey .

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKey Constructor (Single, Single)

Initializes a new instance of [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKey (  
    float position,  
    float value  
)
```

Parameters

position

Position in the curve.

value

Value of the control point.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey Constructor (Single, Single, Single, Single)

Initializes a new instance of [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKey (  
    float position,  
    float value,  
    float tangentIn,  
    float tangentOut  
)
```

Parameters

position

Position in the curve.

value

Value of the control point.

tangentIn

Tangent approaching point from the previous point in the curve.

tangentOut

Tangent leaving point toward next point in the curve.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey Constructor (Single, Single, Single, Single, CurveContinuity)

Initializes a new instance of [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKey (  
    float position,  
    float value,  
    float tangentIn,  
    float tangentOut,  
    CurveContinuity continuity  
)
```

Parameters

position

Position in the curve.

value

Value of the control point.

tangentIn

Tangent approaching point from the previous point in the curve.

tangentOut

Tangent leaving point toward next point in the curve.

continuity

Enum indicating whether the curve is discrete or continuous.

See Also

Reference

[CurveKey Class](#)












[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey Methods

Public Methods

	Name	Description
	Clone	Creates a copy of the CurveKey .
	CompareTo	Compares this instance to another CurveKey and returns an indication of their relative values.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Returns the hash code for this instance.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether two CurveKey instances are equal.
 	op_Inequality	Determines whether two CurveKey instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CurveKey Class](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKey.Clone Method

Creates a copy of the [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKey Clone ()
```

Return Value

The copy of the [CurveKey](#).

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.CompareTo Method

Compares this instance to another [CurveKey](#) and returns an indication of their relative values.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int CompareTo (  
    CurveKey other  
)
```

Parameters

other

[CurveKey](#) to compare to.

Return Value

Zero if the positions are the same; -1 if this [CurveKey](#) comes before **other** and 1 if this [CurveKey](#) comes after **other**.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
CurveKey.Equals (CurveKey)	Determines whether the specified Object is equal to the CurveKey .
CurveKey.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
CurveKey.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKey.Equals Method (CurveKey)

Determines whether the specified [Object](#) is equal to the [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    CurveKey other  
)
```

Parameters

other

The [Object](#) to compare with the current [CurveKey](#).

Return Value

true if the specified [Object](#) is equal to the current [CurveKey](#); **false** otherwise.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

Value that is **true** if the current instance is equal to the specified object, or **false** if it is not.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.GetHashCode Method

Returns the hash code for this instance.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hashcode for this object.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.op_Equality Method

Determines whether two [CurveKey](#) instances are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    CurveKey a,  
    CurveKey b  
)
```

Parameters

a

[CurveKey](#) on the left of the equal sign.

b

[CurveKey](#) on the right of the equal sign.

Return Value

true if the [CurveKeys](#) are equal; **false** otherwise.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.op_Inequality Method

Determines whether two [CurveKey](#) instances are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    CurveKey a,  
    CurveKey b  
)
```

Parameters

a

[CurveKey](#) on the left of the equal sign.

b

[CurveKey](#) on the right of the equal sign.

Return Value

true /> if the [CurveKeys](#) are not equal; **false** otherwise.

See Also

Reference

[CurveKey Class](#)






[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey Properties

Public Properties

	Name	Description
	Continuity	Describes whether the segment between this point and the next point in the curve is discrete or continuous.
	Position	Position of the CurveKey in the curve.
	TangentIn	Describes the tangent when approaching this point from the previous point in the curve.
	TangentOut	Describes the tangent when leaving this point to the next point in the curve.
	Value	Describes the value of this point.

See Also

Reference

[CurveKey Class](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKey.Continuity Property

Describes whether the segment between this point and the next point in the curve is discrete or continuous.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveContinuity Continuity { get; set; }
```

Property Value

Describes the continuity between this point and the next.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.Position Property

Position of the [CurveKey](#) in the curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Position { get; }
```

Property Value

Position of the [CurveKey](#) in the curve.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.TangentIn Property

Describes the tangent when approaching this point from the previous point in the curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float TangentIn { get; set; }
```

Property Value

The tangent when approaching.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.TangentOut Property

Describes the tangent when leaving this point to the next point in the curve.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float TangentOut { get; set; }
```

Property Value

The tangent when leaving.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKey.Value Property

Describes the value of this point.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Value { get; set; }
```

Property Value

The value of the point.

See Also

Reference

[CurveKey Class](#)

[CurveKey Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection Class

Contains the [CurveKeys](#) making up a [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute(System.ComponentModel.ExpandableObjectConverter, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089)]
[SerializableAttribute]
public class CurveKeyCollection : ICollection<CurveKey>, IEnumerable<CurveKey>, IEnumerable
```

See Also

Reference

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)


[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




CurveKeyCollection Members

The following tables list the members exposed by the CurveKeyCollection type.















Public Constructors

Name	Description
 CurveKeyCollection	Initializes a new instance of CurveKeyCollection .



Public Properties

Name	Description
 Count	Gets the number of elements contained in the CurveKeyCollection .
 IsReadOnly	Returns a value indicating whether the CurveKeyCollection is read-only.
 Item	Gets or sets the element at the specified index.


Public Methods

Name	Description
 Add	Adds a CurveKey to the CurveKeyCollection .
 Clear	Removes all CurveKeys from the CurveKeyCollection .
 Clone	Creates a copy of the CurveKeyCollection .
 Contains	Determines whether the CurveKeyCollection contains a specific CurveKey .
 CopyTo	Copies the CurveKeys of the CurveKeyCollection to an array, starting at the array index provided.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the CurveKeyCollection .
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Determines the index of a CurveKey in the CurveKeyCollection .
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the first occurrence of a specific CurveKey from the CurveKeyCollection .
 RemoveAt	Removes the CurveKey at the specified index.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the CurveKeyCollection .

See Also

Reference

[CurveKeyCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKeyCollection Constructor

Initializes a new instance of [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKeyCollection ()
```

See Also

Reference

[CurveKeyCollection Class](#)















[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection Methods


Public Methods

Name	Description
 Add	Adds a CurveKey to the CurveKeyCollection .
 Clear	Removes all CurveKeys from the CurveKeyCollection .
 Clone	Creates a copy of the CurveKeyCollection .
 Contains	Determines whether the CurveKeyCollection contains a specific CurveKey .
 CopyTo	Copies the CurveKeys of the CurveKeyCollection to an array, starting at the array index provided.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the CurveKeyCollection .
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Determines the index of a CurveKey in the CurveKeyCollection .
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the first occurrence of a specific CurveKey from the CurveKeyCollection .
 RemoveAt	Removes the CurveKey at the specified index.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the CurveKeyCollection .

See Also

Reference

[CurveKeyCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKeyCollection.Add Method

Adds a [CurveKey](#) to the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Add (  
    CurveKey item  
)
```

Parameters

item

The [CurveKey](#) to add.

Exceptions

Exception type	Condition
ArgumentNullException	<i>item</i> is null .

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.Clear Method

Removes all [CurveKeys](#) from the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear ()
```

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.Clone Method

Creates a copy of the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKeyCollection Clone ()
```

Return Value

A new object that is a copy of this instance.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.Contains Method

Determines whether the [CurveKeyCollection](#) contains a specific [CurveKey](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Contains (  
    CurveKey item  
)
```

Parameters

item

true if the [CurveKey](#) is found in the [CurveKeyCollection](#); **false** otherwise.

Return Value

The [CurveKey](#) to locate in the [CurveKeyCollection](#).

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.CopyTo Method

Copies the [CurveKeys](#) of the [CurveKeyCollection](#) to an array, starting at the array index provided.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void CopyTo (  
    CurveKey[] array,  
    int arrayIndex  
)
```

Parameters

array

The destination of the [CurveKeys](#) copied from [CurveKeyCollection](#). The array must have zero-based indexing.

arrayIndex

The zero-based index in the array to start copying from.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.GetEnumerator Method

Returns an enumerator that iterates through the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<CurveKey> GetEnumerator ()
```

Return Value

An enumerator for the [CurveKeyCollection](#).

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.IndexOf Method

Determines the index of a [CurveKey](#) in the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int IndexOf (
    CurveKey item
)
```

Parameters

item

[CurveKey](#) to locate in the [CurveKeyCollection](#).

Return Value

The index of the [CurveKey](#) if found in the [CurveKeyCollection](#); -1 otherwise.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.Remove Method

Removes the first occurrence of a specific [CurveKey](#) from the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Remove (  
    CurveKey item  
)
```

Parameters

item

The [CurveKey](#) to remove from the [CurveKeyCollection](#).

Return Value

true if [CurveKey](#) is successfully removed; **false** otherwise.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.RemoveAt Method

Removes the [CurveKey](#) at the specified index.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void RemoveAt (  
    int index  
)
```

Parameters

index

The zero-based index of the item to remove.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

An enumerator for the [CurveKeyCollection](#).

See Also

Reference

[CurveKeyCollection Class](#)




[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of elements contained in the CurveKeyCollection .
	IsReadOnly	Returns a value indicating whether the CurveKeyCollection is read-only.
	Item	Gets or sets the element at the specified index.

See Also

Reference

[CurveKeyCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

CurveKeyCollection.Count Property

Gets the number of elements contained in the [CurveKeyCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of elements in the [CurveKeyCollection](#).

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.IsReadOnly Property

Returns a value indicating whether the [CurveKeyCollection](#) is read-only.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

true if the [CurveKeyCollection](#) is read-only; **false** otherwise.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveKeyCollection.Item Property

Gets or sets the element at the specified index.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CurveKey this [
    int index
] { get; set; }
```

Property Value

The [CurveKey](#) at the specified index.

Exceptions

Exception type	Condition
ArgumentNullException	Item is null.

See Also

Reference

[CurveKeyCollection Class](#)

[CurveKeyCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveLoopType Enumeration

Defines how the value of a [Curve](#) will be determined for positions before the first point on the [Curve](#) or after the last point on the [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CurveLoopType
```

Members

Member name	Description
Constant	The Curve will evaluate to its first key for positions before the first point in the Curve and to the last key for positions after the last point.
Cycle	Positions specified past the ends of the curve will wrap around to the opposite side of the Curve .
CycleOffset	Positions specified past the ends of the curve will wrap around to the opposite side of the Curve . The value will be offset by the difference between the values of the first and last CurveKey multiplied by the number of times the position wraps around. If the position is before the first point in the Curve , the difference will be subtracted from its value; otherwise, the difference will be added.
Linear	Linear interpolation will be performed to determine the value.
Oscillate	Positions specified past the ends of the Curve act as an offset from the same side of the Curve toward the opposite side.

See Also

Reference

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)

[How To: Script the Camera to Follow a Curve](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CurveTangent Enumeration

Specifies different tangent types to be calculated for [CurveKey](#) points in a [Curve](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CurveTangent
```

Members

Member name	Description
Flat	A Flat tangent always has a value equal to zero.
Linear	<p>A Linear tangent at a CurveKey is equal to the difference between its Value and the Value of the preceding or succeeding CurveKey.</p> <p>For example, in Curve MyCurve, where <i>i</i> is greater than zero and (<i>i</i> + 1) is less than the total number of CurveKeys in MyCurve, the linear TangentIn of MyCurve.Keys[i] is equal to:</p> $(\text{MyCurve.Keys}[i].\text{Value} - \text{MyCurve.Keys}[i - 1].\text{Value})$ <p>Similarly, the linear TangentOut is equal to:</p> $(\text{MyCurve.Keys}[i + 1].\text{Value} - \text{MyCurve.Keys}[i].\text{Value})$
Smooth	<p>A Smooth tangent smooths the inflection between a TangentIn and TangentOut by taking into account the values of both neighbors of the CurveKey.</p> <p>The smooth TangentIn of MyCurve.Keys[i] is equal to:</p> $((\text{MyCurve.Keys}[i + 1].\text{Value} - \text{MyCurve.Keys}[i - 1].\text{Value}) * ((\text{MyCurve.Keys}[i].\text{Position} - \text{MyCurve.Keys}[i - 1].\text{Position}) / (\text{MyCurve.Keys}[i + 1].\text{Position} - \text{MyCurve.Keys}[i - 1].\text{Position})))$ <p>Similarly, the smooth TangentOut is equal to:</p> $((\text{MyCurve.Keys}[i + 1].\text{Value} - \text{MyCurve.Keys}[i - 1].\text{Value}) * ((\text{MyCurve.Keys}[i + 1].\text{Position} - \text{MyCurve.Keys}[i].\text{Position}) / (\text{MyCurve.Keys}[i + 1].\text{Position} - \text{MyCurve.Keys}[i - 1].\text{Position})))$

See Also

Reference

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent Class

A game component that is notified when it needs to draw itself.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class DrawableGameComponent : GameComponent, IDrawable
```

See Also

Reference

[DrawableGameComponent Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







DrawableGameComponent Members

The following tables list the members exposed by the `DrawableGameComponent` type.










Public Constructors

Name	Description
 <code>DrawableGameComponent</code>	Creates a new instance of <code>DrawableGameComponent</code> .










Public Properties

Name	Description
 <code>DrawOrder</code>	Order in which the component should be drawn, relative to other components that are in the same <code>GameComponentCollection</code> .
 <code>Enabled</code>	(Inherited from <code>GameComponent</code> .)
 <code>Game</code>	(Inherited from <code>GameComponent</code> .)
 <code>GraphicsDevice</code>	The <code>GraphicsDevice</code> the <code>DrawableGameComponent</code> is associated with.
 <code>UpdateOrder</code>	(Inherited from <code>GameComponent</code> .)
 <code>Visible</code>	Indicates whether <code>Draw</code> should be called.



Public Methods




Name	Description
 <code>Dispose</code>	Overloaded. Releases the resources used by the <code>DrawableGameComponent</code> class.
 <code>Draw</code>	Called when the <code>DrawableGameComponent</code> needs to be drawn. Override this method with component-specific drawing code.
 <code>Equals</code>	(Inherited from <code>Object</code> .)
 <code>GetHashCode</code>	(Inherited from <code>Object</code> .)
 <code>GetType</code>	(Inherited from <code>Object</code> .)
 <code>Initialize</code>	Initializes the component. Override this method to load any non-graphics resources and query for any required services.
 <code>ReferenceEquals</code>	(Inherited from <code>Object</code> .)
 <code>ToString</code>	(Inherited from <code>Object</code> .)
 <code>Update</code>	(Inherited from <code>GameComponent</code> .)

Protected Methods

Name	Description
 <code>LoadContent</code>	Called when graphics resources need to be loaded. Override this method to load any component-specific graphics resources.
 <code>LoadGraphicsContent</code>	Obsolete. Called when the component needs to load graphics resources. Override this method to load any component-specific graphics resources.
 <code>MemberwiseClone</code>	(Inherited from <code>Object</code> .)
 <code>OnDrawOrderChanged</code>	Called when the <code>DrawOrder</code> property changes. Raises the <code>DrawOrderChanged</code> event.
 <code>OnEnabledChanged</code>	(Inherited from <code>GameComponent</code> .)
 <code>OnUpdateOrderChanged</code>	(Inherited from <code>GameComponent</code> .)
 <code>OnVisibleChanged</code>	Called when the <code>Visible</code> property changes. Raises the <code>VisibleChanged</code> event.
 <code>UnloadContent</code>	Called when graphics resources need to be unloaded. Override this method to unload any component-specific graphics resources.
 <code>UnloadGraphicsContent</code>	Obsolete. Called when graphics resources should be unloaded. Override this method to handle component-specific graphics resources.

Public Events

Name	Description
 <code>Disposed</code>	(Inherited from <code>GameComponent</code> .)
 <code>DrawOrderChanged</code>	Raised when the <code>DrawOrder</code> property changes.

 EnabledChanged	(Inherited from GameComponent .)
 UpdateOrderChanged	(Inherited from GameComponent .)
 VisibleChanged	Raised when the Visible property changes.

See Also

Reference

[DrawableGameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

DrawableGameComponent Constructor

Creates a new instance of **DrawableGameComponent**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public DrawableGameComponent (  
    Game game  
)
```

Parameters

game

The [Game](#) that the game component should be attached to.

See Also

Reference

[DrawableGameComponent Class](#)










[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)










Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Releases the resources used by the DrawableGameComponent class.
	Draw	Called when the DrawableGameComponent needs to be drawn. Override this method with component-specific drawing code.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Initialize	Initializes the component. Override this method to load any non-graphics resources and query for any required services.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Update	(Inherited from GameComponent .)

Protected Methods

	Name	Description
	LoadContent	Called when graphics resources need to be loaded. Override this method to load any component-specific graphics resources.
	LoadGraphicsContent	Obsolete. Called when the component needs to load graphics resources. Override this method to load any component-specific graphics resources.
	MemberwiseClone	(Inherited from Object .)
	OnDrawOrderChanged	Called when the DrawOrder property changes. Raises the DrawOrderChanged event.
	OnEnabledChanged	(Inherited from GameComponent .)
	OnUpdateOrderChanged	(Inherited from GameComponent .)
	OnVisibleChanged	Called when the Visible property changes. Raises the VisibleChanged event.
	UnloadContent	Called when graphics resources need to be unloaded. Override this method to unload any component-specific graphics resources.
	UnloadGraphicsContent	Obsolete. Called when graphics resources should be unloaded. Override this method to handle component-specific graphics resources.

See Also

Reference

[DrawableGameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

DrawableGameComponent.Dispose Method

Releases the resources used by the [DrawableGameComponent](#) class.

Overload List

Name	Description
DrawableGameComponent.Dispose (Boolean)	Releases the unmanaged resources used by the DrawableGameComponent and optionally releases the managed resources.
DrawableGameComponent.Dispose ()	(Inherited from GameComponent .)

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

DrawableGameComponent.Dispose Method (Boolean)

Releases the unmanaged resources used by the [DrawableGameComponent](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected override void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.Draw Method

Called when the [DrawableGameComponent](#) needs to be drawn. Override this method with component-specific drawing code.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public virtual void Draw (  
    gameTime  
)
```

Parameters

gameTime

Time passed since the last call to **Draw**.

Remarks

⚠Caution

In the **Draw** method of an Xbox 360 game, vertex buffers, index buffers, and textures should not be written using **SetData** when they are intended to be used for rendering. This condition may lead to graphics corruption or crashes. To avoid this potential issue, use [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#) as the preferred alternative to [VertexBuffer.SetData](#) for dynamic vertex generation.

There is a good reason why you should use [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#). In cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is used on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the **Draw** function are not submitted until [Present](#) is called. (Note that **Draw** implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.Initialize Method

Initializes the component. Override this method to load any non-graphics resources and query for any required services.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public override void Initialize ()
```

Exceptions

Exception type	Condition
InvalidOperationException	There is no IGraphicsDeviceService available in the GameServiceContainer that this DrawableGameComponent is associated with. Usually this means there is no GraphicsDeviceManager associated with the services of a Game .

Remarks

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, [Game.Services](#). Service providers register with [Game.Services](#), and service consumers request services from [Game.Services](#). This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with [Game.Services](#). A service is registered by calling [Game.Services.AddService](#) specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface [IMyService](#), you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by [Game.Services.GetService](#) and specifying the desired service. For example, to retrieve [IGraphicsDeviceService](#), you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Services.GetService( typeof(IGraphicsDeviceService) );
```

See Also

Reference

[Game.Services](#) Property

[DrawableGameComponent](#) Class

[DrawableGameComponent](#) Members

[Microsoft.Xna.Framework](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.LoadContent Method

Called when graphics resources need to be loaded. Override this method to load any component-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void LoadContent ()
```

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.LoadGraphicsContent Method

Obsolete. Called when the component needs to load graphics resources. Override this method to load any component-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void LoadGraphicsContent (
    bool loadAllContent
)
```

Parameters

loadAllContent

true if all graphics resources need to be loaded; **false** if only manual resources need to be loaded.

Remarks

Note

This method is obsolete in XNA Game Studio 2.0. Override [LoadContent](#) to load graphics resources.

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.OnDrawOrderChanged Method

Called when the [DrawOrder](#) property changes. Raises the [DrawOrderChanged](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnDrawOrderChanged (
    Object sender,
    EventArgs args
)
```

Parameters

sender

The [DrawableGameComponent](#).

args

Arguments to the [DrawOrderChanged](#) event.

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.OnVisibleChanged Method

Called when the [Visible](#) property changes. Raises the [VisibleChanged](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnVisibleChanged (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [DrawableGameComponent](#).

args

Arguments to the [VisibleChanged](#) event.

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.UnloadContent Method

Called when graphics resources need to be unloaded. Override this method to unload any component-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void UnloadContent ()
```

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.UnloadGraphicsContent Method

Obsolete. Called when graphics resources should be unloaded. Override this method to handle component-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void UnloadGraphicsContent (  
    bool unloadAllContent  
)
```

Parameters

unloadAllContent

true if all graphics resources should be unloaded; **false** if only manual resources should be unloaded.

Remarks

Note

This method is obsolete in XNA Game Studio 2.0. Override [LoadContent](#) to unload graphics resources.

See Also

Reference

[DrawableGameComponent Class](#)







[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent Properties

Public Properties

	Name	Description
	DrawOrder	Order in which the component should be drawn, relative to other components that are in the same GameComponentCollection .
	Enabled	(Inherited from GameComponent .)
	Game	(Inherited from GameComponent .)
	GraphicsDevice	The GraphicsDevice the DrawableGameComponent is associated with.
	UpdateOrder	(Inherited from GameComponent .)
	Visible	Indicates whether Draw should be called.

See Also

Reference

[DrawableGameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

DrawableGameComponent.DrawOrder Property

Order in which the component should be drawn, relative to other components that are in the same [GameComponentCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public int DrawOrder { get; set; }
```

Property Value

Order in which the component should be drawn.

Remarks This value can be any integer. Components in the [GameComponentCollection](#) are drawn in ascending order based on their **DrawOrder**.

Example

In this example, the three components are added to a [GameComponentCollection](#), and the **DrawOrder** of each component is explicitly set to control the drawing order of each component regardless of the order it was added to the [GameComponentCollection](#). **MyDrawableGameComponent** is a class which derives from [DrawableGameComponent](#), and **game** is an instance of [Game](#).

```
MyDrawableGameComponent thirdGameComponent = new MyDrawableGameComponent(game);  
d3.DrawOrder = 3;  
game.Components.Add(d3);
```

```
MyDrawableGameComponent firstGameComponent = new MyDrawableGameComponent(game);  
d1.DrawOrder = 1;  
game.Components.Add(d1);
```

```
MyDrawableGameComponent secondGameComponent = new MyDrawableGameComponent(game);  
d2.DrawOrder = 2;  
game.Components.Add(d2);
```

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.GraphicsDevice Property

The [GraphicsDevice](#) the [DrawableGameComponent](#) is associated with.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

[GraphicsDevice](#) the [DrawableGameComponent](#) is associated with.

Exceptions

Exception type	Condition
InvalidOperationException	The GraphicsDevice property cannot be used before Initialize has been called.

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.Visible Property

Indicates whether [Draw](#) should be called.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool Visible { get; set; }
```

Property Value

true [Draw](#) should be called; **false** otherwise.

See Also

Reference

[DrawableGameComponent Class](#)






[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent Events

Public Events

	Name	Description
	Disposed	(Inherited from GameComponent .)
	DrawOrderChanged	Raised when the DrawOrder property changes.
	EnabledChanged	(Inherited from GameComponent .)
	UpdateOrderChanged	(Inherited from GameComponent .)
	VisibleChanged	Raised when the Visible property changes.

See Also

Reference

[DrawableGameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

DrawableGameComponent.DrawOrderChanged Event

Raised when the [DrawOrder](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler DrawOrderChanged
```

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DrawableGameComponent.VisibleChanged Event

Raised when the [Visible](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler VisibleChanged
```

See Also

Reference

[DrawableGameComponent Class](#)

[DrawableGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FrameworkDispatcher Class

Implements the Windows-specific portion of a **FrameworkDispatcher** class.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class FrameworkDispatcher
```

See Also

Reference

[FrameworkDispatcher Members](#)








[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



FrameworkDispatcher Members

The following tables list the members exposed by the FrameworkDispatcher type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
 	Update	Updates the status of various framework components (such as power state and media) and raises related events. If your game does not use the Game class, you must call this method (the recommendation is once per frame) yourself.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[FrameworkDispatcher Class](#)



[Microsoft.Xna.Framework Namespace](#)

FrameworkDispatcher Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
 	Update	Updates the status of various framework components (such as power state and media) and raises related events. If your game does not use the Game class, you must call this method (the recommendation is once per frame) yourself.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FrameworkDispatcher Class](#)

[Microsoft.Xna.Framework Namespace](#)

FrameworkDispatcher.Update Method

Updates the status of various framework components (such as power state and media) and raises related events. If your game does not use the [Game](#) class, you must call this method (the recommendation is once per frame) yourself.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Update ()
```

See Also

Reference

[FrameworkDispatcher Class](#)

[FrameworkDispatcher Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game Class

Provides basic graphics device initialization, game logic, and rendering code.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class Game : IDisposable
```

See Also

Tasks

[Your First Game: Microsoft XNA Game Studio in 2D](#)

Programming Guide

[Application Model Overview](#)

Reference

[Game Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune











Game Members

The following tables list the members exposed by the Game type.












Public Constructors

Name	Description
 Game	Initializes a new instance of this class, which provides basic graphics device initialization, game logic, rendering code, and a game loop.













Public Properties






Name	Description
 Components	Gets the collection of GameComponents owned by the game.
 Content	Gets or sets the current ContentManager .
 GraphicsDevice	Gets the current GraphicsDevice .
 InactiveSleepTime	Gets or sets the time to sleep when the game is inactive.
 IsActive	Indicates whether the game is currently the active application.
 IsFixedTimeStep	Gets or sets a value indicating whether to use fixed time steps.
 IsMouseVisible	Gets or sets a value indicating whether the mouse cursor should be visible.
 Services	Gets the GameServiceContainer holding all the service providers attached to the Game .
 TargetElapsedTime	Gets or sets the target time between calls to Update when IsFixedTimeStep is true.
 Window	Gets the underlying operating system window.

Public Methods





Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 Exit	Exits the game.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ResetElapsedTime	Resets the elapsed time counter.
 Run	Call this method to initialize the game, begin running the game loop, and start processing events for the game.
 SuppressDraw	Prevents calls to Draw until the next Update .
 Tick	Updates the game's clock and calls Update and Draw .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 BeginDraw	Starts the drawing of a frame. This method is followed by calls to Draw and EndDraw .
 BeginRun	Called after all components are initialized but before the first update in the game loop.
 Draw	Called when the game determines it is time to draw a frame.
 EndDraw	Ends the drawing of a frame. This method is preceded by calls to Draw and BeginDraw .
 EndRun	Called after the game loop has stopped running before exiting.
 Finalize	Allows a Game to attempt to free resources and perform other cleanup operations before garbage collection reclaims the Game .
 Initialize	Called after the Game and GraphicsDevice are created, but before LoadContent .
 LoadContent	Called when graphics resources need to be loaded.
 LoadGraphicsContent	Obsolete. Called when graphics resources need to be loaded. Override this method to load any game-specific graphics resources.
 MemberwiseClone	(Inherited from Object .)
 OnActivated	Raises the Activated event. Override this method to add code to handle when the game gains focus.
 OnDeactivated	Raises the Deactivated event. Override this method to add code to handle when the game loses focus.

 OnExiting	Raises an Exiting event. Override this method to add code to handle when the game is exiting.
 ShowMissingRequirementMessage	This is used to display an error message if there is no suitable graphics device or sound card.
 UnloadContent	Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.
 UnloadGraphicsContent	Obsolete. Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.
 Update	Called when the game has determined that game logic needs to be processed.

Public Events

	Name	Description
	Activated	Raised when the game gains focus.
	Deactivated	Raised when the game loses focus.
	Disposed	Raised when the game is being disposed.
	Exiting	Raised when the game is exiting.

See Also

Reference

[Game Class](#)

[Microsoft.Xna.Framework Namespace](#)

Game Constructor

Initializes a new instance of this class, which provides basic graphics device initialization, game logic, rendering code, and a game loop.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public Game ()
```

Remarks

Although the [GraphicsDeviceManager](#) constructor requires an instance of [Game](#), if the application does not otherwise require an instance of [Game](#) it is often more desirable to implement the [IGraphicsDeviceService](#) interface to provide the same services that would be provided by the [GraphicsDeviceManager](#).

Similarly, when creating a new [ContentManager](#), if no instance of [Game](#) is otherwise required by the application, it is often more desirable to create a new class that implements the [IServiceProvider](#) interface rather than creating an instance of [Game](#) just to create a new instance of [GraphicsDeviceManager](#).

Example

```
/// <summary>
/// The main entry point for the application.
/// </summary>
static void Main(string[] args)
{
    using (Game1 game = new Game1())
    {
        game.Run();
    }
}
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game Methods

Public Methods

Name	Description
Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
Equals	(Inherited from Object .)
Exit	Exits the game.
GetHashCode	(Inherited from Object .)
GetType	(Inherited from Object .)
ReferenceEquals	(Inherited from Object .)
ResetElapsedTime	Resets the elapsed time counter.
Run	Call this method to initialize the game, begin running the game loop, and start processing events for the game.
SuppressDraw	Prevents calls to Draw until the next Update .
Tick	Updates the game's clock and calls Update and Draw .
ToString	(Inherited from Object .)

Protected Methods

Name	Description
BeginDraw	Starts the drawing of a frame. This method is followed by calls to Draw and EndDraw .
BeginRun	Called after all components are initialized but before the first update in the game loop.
Draw	Called when the game determines it is time to draw a frame.
EndDraw	Ends the drawing of a frame. This method is preceded by calls to Draw and BeginDraw .
EndRun	Called after the game loop has stopped running before exiting.
Finalize	Allows a Game to attempt to free resources and perform other cleanup operations before garbage collection reclaims the Game .
Initialize	Called after the Game and GraphicsDevice are created, but before LoadContent .
LoadContent	Called when graphics resources need to be loaded.
LoadGraphicsContent	Obsolete. Called when graphics resources need to be loaded. Override this method to load any game-specific graphics resources.
MemberwiseClone	(Inherited from Object .)
OnActivated	Raises the Activated event. Override this method to add code to handle when the game gains focus.
OnDeactivated	Raises the Deactivated event. Override this method to add code to handle when the game loses focus.
OnExiting	Raises an Exiting event. Override this method to add code to handle when the game is exiting.
ShowMissingRequirementMessage	This is used to display an error message if there is no suitable graphics device or sound card.
UnloadContent	Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.
UnloadGraphicsContent	Obsolete. Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.
Update	Called when the game has determined that game logic needs to be processed.

See Also

Reference

[Game Class](#)

[Microsoft.Xna.Framework Namespace](#)

Game.BeginDraw Method

Starts the drawing of a frame. This method is followed by calls to [Draw](#) and [EndDraw](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual bool BeginDraw ()
```

Return Value

true if the frame should be drawn; **false** otherwise.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.BeginRun Method

Called after all components are initialized but before the first update in the game loop.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void BeginRun ()
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
Game.Dispose ()	Immediately releases the unmanaged resources used by this object.
Game.Dispose (Boolean)	Releases all resources used by the Game class.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Game.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Dispose Method (Boolean)

Releases all resources used by the [Game](#) class.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [Game](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Draw Method

Called when the game determines it is time to draw a frame. Override this method with game-specific rendering code.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void Draw (
    gameTime
```

Parameters

gameTime

Time passed since the last call to **Draw**.

Remarks

[Update](#) and [Draw](#) are called at different rates depending on whether [IsFixedTimeStep](#) is **true** or **false**. If [IsFixedTimeStep](#) is **false**, [Update](#) and [Draw](#) will be called sequentially as often as possible. If [IsFixedTimeStep](#) is **true**, [Update](#) will be called at the interval specified in [TargetElapsedTime](#), while [Draw](#) will continue to be called as often as possible. For more information on fixed-step and variable-step game loops, see [Application Model Overview](#).

⚠Caution

In the **Draw** method of an Xbox 360 game, vertex buffers, index buffers, and textures should not be written using **SetData** when they are intended to be used for rendering. This condition may lead to graphics corruption or crashes. To avoid this potential issue, use [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#) as the preferred alternative to [VertexBuffer.SetData](#) for dynamic vertex generation.

There is a good reason why you should use [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#). In cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is used on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the **Draw** function are not submitted until [Present](#) is called. (Note that **Draw** implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

Example

In classes that derive from [Game](#), it is necessary to make these calls:

- Call base.**Draw** in **Draw** to enumerate through any graphics components that have been added to [Components](#). This method will automatically call the [Initialize](#) method for every component that has been added to the collection.

```
protected override void Draw( gameTime )
{
    base.Draw( gameTime );
}
```

See Also

Tasks

[Application Model Overview](#)

Concepts

[Game State Management Sample at XNA Creators Club Online](#)

[Predicated Tiling](#)

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.EndDraw Method

Ends the drawing of a frame. This method is preceded by calls to [Draw](#) and [BeginDraw](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void EndDraw ()
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.EndRun Method

Called after the game loop has stopped running before exiting.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void EndRun ()
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Exit Method

Exits the game.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Exit ()
```

Exceptions

Exception type	Condition
InvalidOperationException	The current platform does not allow games to exit.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Exit a Game](#)

[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Finalize Method

Allows a [Game](#) to attempt to free resources and perform other cleanup operations before garbage collection reclaims the [Game](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Initialize Method

Called after the [Game](#) and [GraphicsDevice](#) are created, but before [LoadContent](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void Initialize ()
```

Remarks

Override this method to query for any required services, and load any non-graphics resources. Use [LoadContent](#) to load graphics resources.

Initialize is called before [Draw](#), so the length of time spent executing code in this method will be experienced by the user as a delay before he or she sees the initial game screen.

Example

In classes that derive from [Game](#), you need to call base.**Initialize** in **Initialize**, which will automatically enumerate through any game components that have been added to [Game.Components](#) and call their [Initialize](#) methods.

```
protected override void Initialize()  
{  
    base.Initialize();  
}
```

See Also

Reference

[Game Class](#)

[Game.LoadContent Method](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.LoadContent Method

Called when graphics resources need to be loaded. Override this method to load any game-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void LoadContent ()
```

Remarks

This method is called by [Initialize](#). Also, it is called any time the game content needs to be reloaded, such as when the [DeviceReset](#) event occurs. You should not access the [GraphicsDevice](#) until [LoadContent](#) is called.

LoadContent is called by [Initialize](#), and before [Draw](#). During the time the code for this method is executing, the user will experience a delay before the initial game screen appears.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.LoadGraphicsContent Method

Obsolete. Called when graphics resources need to be loaded. Override this method to load any game-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void LoadGraphicsContent (
    bool loadAllContent
)
```

Parameters

loadAllContent

true if all graphics resources need to be loaded; **false** if only manual resources need to be loaded.

Remarks

Note

This method is obsolete in XNA Game Studio 2.0. Override [LoadContent](#) to load graphics resources.

This method is called by [Initialize](#), as well as any time the game content needs to be reloaded, such as when the [DeviceReset](#) event occurs.

See Also

Tasks

[How To: Load Content](#)

Reference

[Game Class](#)

[LoadContent](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.OnActivated Method

Raises the [Activated](#) event. Override this method to add code to handle when the game gains focus.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnActivated (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [Game](#).

args

Arguments for the [Activated](#) event.

Remarks The **OnActivated** method allows derived classes to handle the [Activated](#) event without attaching a delegate. This is the preferred technique for handling the event in a derived class.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.OnDeactivated Method

Raises the [Deactivated](#) event. Override this method to add code to handle when the game loses focus.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnDeactivated (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [Game](#).

args

Arguments for the [Deactivated](#) event.

Remarks The **OnDeactivated** method allows derived classes to handle the [Deactivated](#) event without attaching a delegate. This is the preferred technique for handling the event in a derived class.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.OnExiting Method

Raises an [Exiting](#) event. Override this method to add code to handle when the game is exiting.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnExiting (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [Game](#).

args

Arguments for the [Exiting](#) event.

Remarks The **OnExiting** method allows derived classes to handle the [Exiting](#) event without attaching a delegate. This is the preferred technique for handling the event in a derived class.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.ResetElapsedTime Method

Resets the elapsed time counter.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ResetElapsedTime ()
```

Remarks

Use this method if your game is recovering from a slow-running state, and the [ElapsedGameTime](#) or [ElapsedRealTime](#) is too large to be useful.

See Also

Reference

[Game Class](#)

[Game Members](#)

[ElapsedGameTime](#)

[ElapsedRealTime](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Run Method

Call this method to initialize the game, begin running the game loop, and start processing events for the game.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Run ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Run has been called multiple times in the same application.

ArgumentException	<p>Invalid arguments have been requested for the creation of the graphics device.</p> <p>Invalid back buffer options have been requested:</p> <ul style="list-style-type: none"> • <code>BackBufferCount</code> is not between 0 and 3. • <code>BackBufferFormat</code> is not valid. Try <code>SurfaceFormat.Bgr565</code>, <code>SurfaceFormat.Bgr555</code>, <code>SurfaceFormat.Bgra5551</code>, <code>SurfaceFormat.Bgr32</code>, <code>SurfaceFormat.Color</code>, <code>SurfaceFormat.Bgra1010102</code>. • The selected <code>BackBufferFormat</code> and <code>IsFullScreen</code> values are not valid for the selected adapter format and device type. • The <code>BackBufferWidth</code> and <code>BackBufferHeight</code> values do not correspond to a valid display mode. • <code>FullScreenRefreshRateInHz</code> does not match a valid display mode for the specified <code>BackBufferWidth</code> and <code>BackBufferHeight</code>. <p>Invalid <code>PresentationParameters.PresentationInterval</code> or <code>SwapEffect</code> options have been requested:</p> <ul style="list-style-type: none"> • On Xbox 360, <code>PresentationParameters.PresentationInterval</code> has been assigned the invalid value <code>PresentInterval.Four</code>. • <code>SwapEffect.Copy</code> has been assigned to the <code>PresentationParameters.SwapEffect</code> property, but <code>BackBufferCount</code> is not 1. When using <code>SwapEffect.Copy</code>, <code>BackBufferCount</code> must be 1. • <code>SwapEffect</code> is not one of the following: <code>SwapEffect.Copy</code>, <code>SwapEffect.Discard</code>, or <code>SwapEffect.Flip</code>. <p>Invalid multisampling options have been requested:</p> <ul style="list-style-type: none"> • Multisampling has been enabled, but <code>SwapEffect.Discard</code> has not been specified for the <code>SwapEffect</code> property. <code>SwapEffect.Discard</code> is required when enabling multisampling. • The selected <code>MultiSampleType</code> is not compatible with the current <code>BackBufferFormat</code> and <code>IsFullScreen</code> value for the selected adapter. • The selected <code>MultiSampleQuality</code> value is invalid for the selected <code>MultiSampleType</code>. <p>Invalid depth stencil options have been requested:</p> <ul style="list-style-type: none"> • <code>AutoDepthStencilFormat</code> is not supported as a depth/stencil format when using the selected adapter <code>BackBufferFormat</code>. • <code>PresentOptions</code> has been set to <code>PresentOptions.DiscardDepthStencil</code>, but <code>EnableAutoDepthStencil</code> is not true. • Multiple <code>PresentationParameters</code> have been specified and <code>EnableAutoDepthStencil</code> is true, but <code>EnableAutoDepthStencil</code>, <code>BackBufferFormat</code>, <code>BackBufferWidth</code>, and <code>BackBufferHeight</code> are not the same for all adapters. <p><code>IsFullScreen</code> is true, but:</p> <ul style="list-style-type: none"> • <code>BackBufferWidth</code> and <code>BackBufferHeight</code> are not nonzero. • <code>PresentationInterval</code> is not one of the following: <code>PresentInterval.Default</code>, <code>PresentInterval.Immediate</code>, <code>PresentInterval.One</code>, <code>PresentInterval.Two</code>, <code>PresentInterval.Three</code>, or <code>PresentInterval.Four</code>. • <code>FullScreenRefreshRateInHz</code> is not nonzero. • The selected adapter does not support the selected full screen <code>PresentationInterval</code>. <p><code>IsFullScreen</code> is false, but:</p> <ul style="list-style-type: none"> • <code>PresentationInterval</code> is not one of the following: <code>PresentInterval.Default</code>, <code>PresentInterval.Immediate</code>, or <code>PresentInterval.One</code>. • <code>FullScreenRefreshRateInHz</code> is not zero.
-------------------	---

Remarks

This method calls the game `Initialize` and `BeginRun` methods before it begins the game loop and starts processing events for the game.

Example

C#

```
static class Program
{
```

```
/// <summary>
/// The main entry point for the application.
/// </summary>
static void Main(string[] args)
{
    using (Game1 game = new Game1())
    {
        game.Run();
    }
}
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.ShowMissingRequirementMessage Method

This is used to display an error message if there is no suitable graphics device or sound card.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual bool ShowMissingRequirementMessage (  
    Exception exception  
)
```

Parameters

exception

The exception to display.

Return Value

true if an error was displayed, otherwise **false**.

Remarks

Xbox 360 Specific Information

On the Xbox 360 platform, this method will not display a message box and return immediately.

Zune Specific Information

On the Zune platform, this method will not display a message box and return immediately.

Games are not expected to call this method directly. This method will be called automatically if the game throws an unhandled exception of type [NoSuitableGraphicsDeviceException](#) or [NoAudioHardwareException](#). You may override this method to add messages for additional exception types, or to provide localized versions of the error message text.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.SuppressDraw Method

Prevents calls to [Draw](#) until the next [Update](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SuppressDraw ()
```

Remarks

Call this method during [Update](#) to prevent any calls to [Draw](#) until after the next call to [Update](#). This method can be used on small devices to conserve battery life if the display does not change as a result of [Update](#). For example, if the screen is static with no background animations, the player input can be examined during [Update](#) to determine whether the player is performing any action. If no input is detected, this method allows the game to skip drawing until the next update.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Tick Method

Updates the game's clock and calls [Update](#) and [Draw](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Tick ()
```

Remarks

In a fixed-step game, [Tick](#) calls [Update](#) only after a target time interval has elapsed.

In a variable-step game, [Update](#) is called every time [Tick](#) is called.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.UnloadContent Method

Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void UnloadContent ()
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.UnloadGraphicsContent Method

Obsolete. Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void UnloadGraphicsContent (
    bool unloadAllContent
)
```

Parameters

unloadAllContent

true if all graphics resources need to be unloaded; **false** if only manual resources need to be unloaded.

Remarks

Note

This method is obsolete in XNA Game Studio 2.0. Use [UnloadContent](#) to unload graphics resources.

See Also

Reference

[Game Class](#)

[UnloadContent](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Update Method

Called when the game has determined that game logic needs to be processed. This might include the management of the game state, the processing of user input, or the updating of simulation data. Override this method with game-specific logic.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void Update (  
    gameTime  
)
```

Parameters

gameTime

Time passed since the last call to **Update**.

Remarks

★ Best Practice

When updating the game state, you can examine [SignedInGamer.GameDefaults](#) to determine a player's preferences for settings such as game difficulty and controller sensitivity.

[Update](#) and [Draw](#) are called at different rates depending on whether [IsFixedTimeStep](#) is **true** or **false**. If [IsFixedTimeStep](#) is **false**, [Update](#) and [Draw](#) will be called in a continuous loop. If [IsFixedTimeStep](#) is **true**, [Update](#) will be called at the interval specified in [TargetElapsedTime](#), while [Draw](#) will only be called if an [Update](#) is not due. If [Draw](#) is not called, [IsRunningSlowly](#) will be set to **true**.

For more information on fixed-step and variable-step game loops, see [Application Model Overview](#).

See Also

Tasks

[How To: Pause a Game](#)

[How To: Make a Game Use a Variable Time Step](#)

[How To: Make a Game Time Out](#)

[How To: Script the Camera to Follow a Curve](#)

Concepts

[Game State Management Sample at XNA Creators Club Online](#)

[Application Model Overview](#)

Reference

[SignedInGamer.GameDefaults](#)

[Game Class](#)











[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game Properties

Public Properties

	Name	Description
	Components	Gets the collection of GameComponents owned by the game.
	Content	Gets or sets the current ContentManager .
	GraphicsDevice	Gets the current GraphicsDevice .
	InactiveSleepTime	Gets or sets the time to sleep when the game is inactive.
	IsActive	Indicates whether the game is currently the active application.
	IsFixedTimeStep	Gets or sets a value indicating whether to use fixed time steps.
	IsMouseVisible	Gets or sets a value indicating whether the mouse cursor should be visible.
	Services	Gets the GameServiceContainer holding all the service providers attached to the Game .
	TargetElapsedTime	Gets or sets the target time between calls to Update when IsFixedTimeStep is true.
	Window	Gets the underlying operating system window.

See Also

Reference

[Game Class](#)

[Microsoft.Xna.Framework Namespace](#)

Game.Components Property

Gets the collection of [GameComponents](#) owned by the game.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameComponentCollection Components { get; }
```

Property Value

The collection of game [GameComponents](#).

Remarks

Game components provide a modular way of adding functionality to a game. You create a game component by deriving the new component either from the [GameComponent](#) class, or, if the component loads and draws graphics content, from the [DrawableGameComponent](#) class. You then add game logic and rendering code to the game component by overriding [GameComponent.Update](#), [DrawableGameComponent.Draw](#) and [GameComponent.Initialize](#). A game component is registered with a game by passing the component to [Game.Components.Add](#). A registered component will have its draw, update, and initialize methods called from the [Game.Initialize](#), [Game.Update](#), and [Game.Draw](#) methods.

See Also

Tasks

[Game Component Development](#)

Concepts

[Application Model Overview](#)

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Content Property

Gets or sets the current [ContentManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentManager Content { get; set; }
```

Property Value

The current [ContentManager](#).

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.GraphicsDevice Property

Gets the current [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The current [GraphicsDevice](#).

Exceptions

Exception type	Condition
InvalidOperationException	This property requires a graphics device service in the game service container.

Remarks

Do not access this property until [LoadContent](#) is called.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.InactiveSleepTime Property

Gets or sets the time to sleep when the game is inactive.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public TimeSpan InactiveSleepTime { get; set; }
```

Property Value

The time to sleep when the game is inactive.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The value specified for <i>InactiveSleepTime</i> is not greater than or equal to zero. Specify zero or a positive value.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.IsActive Property

Indicates whether the game is currently the active application.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool IsActive { get; }
```

Property Value

true if the game is active; **false** otherwise.

Remarks

On Windows, a game is active if it is not minimized and has the current input focus. On Xbox 360, the game is active if the user is not interacting with the Guide or Dashboard.

★Best Practice

It is generally a good idea to pause the game when [Game.IsActive](#) is **false**.

See Also

Tasks

[How To: Pause a Game](#)

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.IsFixedTimeStep Property

Gets or sets a value indicating whether to use fixed time steps.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool IsFixedTimeStep { get; set; }
```

Property Value

true if using fixed time steps; **false** otherwise.

Remarks

The default value for [IsFixedTimeStep](#) is **true**.

A fixed-step [Game](#) tries to call its [Update](#) method on the fixed interval specified in [TargetElapsedTime](#). Setting [Game.IsFixedTimeStep](#) to **true** causes a [Game](#) to use a fixed-step game loop. A new XNA project uses a fixed-step game loop with a default [TargetElapsedTime](#) of 1/60th of a second.

In a fixed-step game loop, [Game](#) calls [Update](#) once the [TargetElapsedTime](#) has elapsed. After [Update](#) is called, if it is not time to call [Update](#) again, [Game](#) calls [Draw](#). After [Draw](#) is called, if it is not time to call [Update](#) again, [Game](#) idles until it is time to call [Update](#).

If [Update](#) takes too long to process, [Game](#) sets [IsRunningSlowly](#) to **true** and calls [Update](#) again, without calling [Draw](#) in between. When an update runs longer than the [TargetElapsedTime](#), [Game](#) responds by calling [Update](#) extra times and dropping the frames associated with those updates to catch up. This ensures that [Update](#) will have been called the expected number of times when the game loop catches up from a slowdown. You can check the value of [IsRunningSlowly](#) in your [Update](#) if you want to detect dropped frames and shorten your [Update](#) processing to compensate. You can reset the elapsed times by calling [ResetElapsedTime](#).

When your game pauses in the debugger, [Game](#) will not make extra calls to [Update](#) when the game resumes.

See Also

Tasks

[How To: Make a Game Use a Variable Time Step](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime.IsRunningSlowly Property](#)

[Game.TargetElapsedTime Property](#)

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.IsMouseVisible Property

Gets or sets a value indicating whether the mouse cursor should be visible.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool IsMouseVisible { get; set; }
```

Property Value

true if the mouse cursor should be visible; **false** otherwise.

Remarks

Note

This property applies only to Windows development. The [Mouse](#) and [MouseState](#) objects are not supported on Xbox 360.

See Also

Tasks

[How To: Detect Whether a User Clicked a 3D Object](#)

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Services Property

Gets the [GameServiceContainer](#) holding all the service providers attached to the [Game](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameServiceContainer Services { get; }
```

Property Value

The [GameServiceContainer](#) holding all the service providers attached to the [Game](#).

Remarks

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, [Game.Services](#). Service providers register with [Game.Services](#), and service consumers request services from [Game.Services](#). This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with [Game.Services](#). A service is registered by calling [Game.Services.AddService](#) specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface `IMyService`, you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by [Game.Services.GetService](#) and specifying the desired service. For example, to retrieve [IGraphicsDeviceService](#), you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Services.GetService( t  
ypeof(IGraphicsDeviceService) );
```

See Also

Concepts

[Application Model Overview](#)

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.TargetElapsedTime Property

Gets or sets the target time between calls to [Update](#) when [IsFixedTimeStep](#) is **true**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public TimeSpan TargetElapsedTime { get; set; }
```

Property Value

The target time period for the game loop.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The value specified for <i>TargetElapsedTime</i> is not greater than zero. Specify a nonzero positive value.

Remarks

When the game framerate is less than **TargetElapsedTime**, [IsRunningSlowly](#) will return **true**.

The default value for [TargetElapsedTime](#) is 1/60th of a second.

A fixed-step [Game](#) tries to call its [Update](#) method on the fixed interval specified in [TargetElapsedTime](#). Setting [Game.IsFixedTimeStep](#) to **true** causes a [Game](#) to use a fixed-step game loop. A new XNA project uses a fixed-step game loop with a default [TargetElapsedTime](#) of 1/60th of a second.

In a fixed-step game loop, [Game](#) calls [Update](#) once the [TargetElapsedTime](#) has elapsed. After [Update](#) is called, if it is not time to call [Update](#) again, [Game](#) calls [Draw](#). After [Draw](#) is called, if it is not time to call [Update](#) again, [Game](#) idles until it is time to call [Update](#).

If [Update](#) takes too long to process, [Game](#) sets [IsRunningSlowly](#) to **true** and calls [Update](#) again, without calling [Draw](#) in between. When an update runs longer than the [TargetElapsedTime](#), [Game](#) responds by calling [Update](#) extra times and dropping the frames associated with those updates to catch up. This ensures that [Update](#) will have been called the expected number of times when the game loop catches up from a slowdown. You can check the value of [IsRunningSlowly](#) in your [Update](#) if you want to detect dropped frames and shorten your [Update](#) processing to compensate. You can reset the elapsed times by calling [ResetElapsedTime](#).

When your game pauses in the debugger, [Game](#) will not make extra calls to [Update](#) when the game resumes.

See Also

Tasks

[How To: Write Games for Less Capable Hardware](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime.IsRunningSlowly Property](#)

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Window Property

Gets the underlying operating system window.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameWindow Window { get; }
```

Property Value

The underlying operating system window.

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Tasks

[How To: Allow the Player to Resize a Game Window](#)

Reference

[Game Class](#)





[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game Events

Public Events

	Name	Description
	Activated	Raised when the game gains focus.
	Deactivated	Raised when the game loses focus.
	Disposed	Raised when the game is being disposed.
	Exiting	Raised when the game is exiting.

See Also

Reference

[Game Class](#)

[Microsoft.Xna.Framework Namespace](#)

Game.Activated Event

Raised when the game gains focus.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler Activated
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Deactivated Event

Raised when the game loses focus.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler Deactivated
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Disposed Event

Raised when the game is being disposed.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler Disposed
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Game.Exiting Event

Raised when the game is exiting.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler Exiting
```

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Exit a Game](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent Class

Base class for all XNA Framework game components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class GameComponent : IGameComponent, IUpdateable, IDisposable
```

Remarks

Game components provide a modular way of adding functionality to a game. You create a game component by deriving the new component either from the [GameComponent](#) class, or, if the component loads and draws graphics content, from the [DrawableGameComponent](#) class. You then add game logic and rendering code to the game component by overriding [GameComponent.Update](#), [DrawableGameComponent.Draw](#) and [GameComponent.Initialize](#). A game component is registered with a game by passing the component to [Game.Components.Add](#). A registered component will have its draw, update, and initialize methods called from the [Game.Initialize](#), [Game.Update](#), and [Game.Draw](#) methods.

See Also

Concepts

[Application Model Overview](#)

Tasks

[Game Component Development](#)

Reference

[GameComponent Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




GameComponent Members

The following tables list the members exposed by the GameComponent type.









Public Constructors

Name	Description
 GameComponent	Initializes a new instance of this class.





Public Properties

Name	Description
 Enabled	Indicates whether GameComponent.Update should be called when Game.Update is called.
 Game	Gets the Game associated with this GameComponent .
 UpdateOrder	Indicates the order in which the GameComponent should be updated relative to other GameComponent instances. Lower values are updated first.




Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Initialize	Called when the GameComponent needs to be initialized.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Update	Called when the GameComponent needs to be updated. Override this method with component-specific update code.

Protected Methods

Name	Description
 Finalize	Allows a GameComponent to attempt to free resources and perform other cleanup operations before garbage collection reclaims the GameComponent .
 MemberwiseClone	(Inherited from Object .)
 OnEnabledChanged	Called when the Enabled property changes. Raises the EnabledChanged event.
 OnUpdateOrderChanged	Called when the UpdateOrder property changes. Raises the UpdateOrderChanged event.

Public Events

Name	Description
 Disposed	Raised when the GameComponent is disposed.
 EnabledChanged	Raised when the Enabled property changes.
 UpdateOrderChanged	Raised when the UpdateOrder property changes.

See Also

Reference

[GameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponent Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameComponent (  
    Game game  
)
```

Parameters

game

Game that the game component should be attached to.

Remarks

Game components provide a modular way of adding functionality to a game. You create a game component by deriving the new component either from the [GameComponent](#) class, or, if the component loads and draws graphics content, from the [DrawableGameComponent](#) class. You then add game logic and rendering code to the game component by overriding [GameComponent.Update](#), [DrawableGameComponent.Draw](#) and [GameComponent.Initialize](#). A game component is registered with a game by passing the component to [Game.Components.Add](#). A registered component will have its draw, update, and initialize methods called from the [Game.Initialize](#), [Game.Update](#), and [Game.Draw](#) methods.

See Also

Tasks

[Game Component Development](#)

Concepts

[Application Model Overview](#)

Reference

[GameComponent Class](#)









[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Initialize	Called when the GameComponent needs to be initialized.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Update	Called when the GameComponent needs to be updated. Override this method with component-specific update code.

Protected Methods

	Name	Description
	Finalize	Allows a GameComponent to attempt to free resources and perform other cleanup operations before garbage collection reclaims the GameComponent .
	MemberwiseClone	(Inherited from Object .)
	OnEnabledChanged	Called when the Enabled property changes. Raises the EnabledChanged event.
	OnUpdateOrderChanged	Called when the UpdateOrder property changes. Raises the UpdateOrderChanged event.

See Also

Reference

[GameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponent.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
GameComponent.Dispose ()	Immediately releases the unmanaged resources used by this object.
GameComponent.Dispose (Boolean)	Releases the unmanaged resources used by the GameComponent and optionally releases the managed resources.

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponent.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.Dispose Method (Boolean)

Releases the unmanaged resources used by the [GameComponent](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.Finalize Method

Allows a [GameComponent](#) to attempt to free resources and perform other cleanup operations before garbage collection reclaims the [GameComponent](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection.

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.Initialize Method

Called when the [GameComponent](#) needs to be initialized. Override this method to query for required services and load any non-graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public virtual void Initialize ()
```

Remarks If this [GameComponent](#) has been added to the [Game.Components](#) collection, this method will be called automatically if you call **base.Initialize** in your [Game.Initialize](#) method. Note that the template code created by XNA Game Studio will already have this call to **base.Initialize** in place.

Example

In classes that derive from [Game](#), it is necessary to call **base.Initialize** in [Game.Initialize](#), which will automatically enumerate through any game components that have been added to [Game.Components](#) and call their [Initialize](#) methods.

```
protected override void Initialize()  
{  
    base.Initialize();  
}
```

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.OnEnabledChanged Method

Called when the [Enabled](#) property changes. Raises the [EnabledChanged](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnEnabledChanged (
    Object sender,
    EventArgs args
)
```

Parameters

sender

The [GameComponent](#).

args

Arguments to the [EnabledChanged](#) event.

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.OnUpdateOrderChanged Method

Called when the [UpdateOrder](#) property changes. Raises the [UpdateOrderChanged](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnUpdateOrderChanged (
    Object sender,
    EventArgs args
)
```

Parameters

sender

The [GameComponent](#).

args

Arguments to the [UpdateOrderChanged](#) event.

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.Update Method

Called when the [GameComponent](#) needs to be updated. Override this method with component-specific update code.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public virtual void Update (  
    gameTime gameTime  
)
```

Parameters

gameTime

Time elapsed since the last call to **Update**

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)




Programming Guide

[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent Properties

Public Properties

	Name	Description
	Enabled	Indicates whether GameComponent.Update should be called when Game.Update is called.
	Game	Gets the Game associated with this GameComponent .
	UpdateOrder	Indicates the order in which the GameComponent should be updated relative to other GameComponent instances. Lower values are updated first.

See Also

Reference

[GameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponent.Enabled Property

Indicates whether [GameComponent.Update](#) should be called when [Game.Update](#) is called.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool Enabled { get; set; }
```

Property Value

true if [GameComponent.Update](#) should be called; **false** otherwise.

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.Game Property

Gets the [Game](#) associated with this [GameComponent](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public Game Game { get; }
```

Property Value

The [Game](#) associated with this [GameComponent](#).

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.UpdateOrder Property

Indicates the order in which the [GameComponent](#) should be updated relative to other [GameComponent](#) instances. Lower values are updated first.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public int UpdateOrder { get; set; }
```

Property Value

The order in which the [GameComponent](#) should be updated.

See Also

Reference

[GameComponent Class](#)




[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent Events

Public Events

	Name	Description
	Disposed	Raised when the GameComponent is disposed.
	EnabledChanged	Raised when the Enabled property changes.
	UpdateOrderChanged	Raised when the UpdateOrder property changes.

See Also

Reference

[GameComponent Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponent.Disposed Event

Raised when the [GameComponent](#) is disposed.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler Disposed
```

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.EnabledChanged Event

Raised when the [Enabled](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler EnabledChanged
```

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponent.UpdateOrderChanged Event

Raised when the [UpdateOrder](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler UpdateOrderChanged
```

See Also

Reference

[GameComponent Class](#)

[GameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection Class

A collection of game components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public sealed class GameComponentCollection : Collection<IGameComponent>
```

See Also

Reference

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



GameComponentCollection Members

The following tables list the members exposed by the GameComponentCollection type.


Public Constructors

Name	Description
 GameComponentCollection	Initializes a new instance of this class.

















Public Properties

Name	Description
 Count	(Inherited from Collection .)
 Item	(Inherited from Collection .)





Protected Properties

Name	Description
 Items	(Inherited from Collection .)



Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 SetItem	Overloaded. Modifies the specified child object in the collection.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	Removes all children from the collection.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes a child object in the collection.

Public Events

Name	Description
 ComponentAdded	Raised when a component is added to the GameComponentCollection .
 ComponentRemoved	Raised when a component is removed from the GameComponentCollection .

See Also

Reference

[GameComponentCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollection Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GameComponentCollection ()
```

See Also

Reference

[GameComponentCollection Class](#)

















[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection Methods

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 SetItem	Overloaded. Modifies the specified child object in the collection.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	Removes all children from the collection.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes a child object in the collection.

See Also

Reference

[GameComponentCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollection.ClearItems Method

Removes all children from the collection.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void ClearItems ()
```

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection.InsertItem Method

Inserts a child object into the collection at the specified location.

Overload List

Name	Description
GameComponentCollection.InsertItem (Int32, IGameComponent)	Inserts a child object into the collection at the specified location.
GameComponentCollection.InsertItem (Int32, IGameComponent)	(Inherited from Collection .)

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollection.InsertItem Method (Int32, IGameComponent)

Inserts a child object into the collection at the specified location.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void InsertItem (  
    int index,  
    IGameComponent item  
)
```

Parameters

index

The position in the collection.

item

The child object being inserted.

Exceptions

Exception type	Condition
ArgumentException	Cannot add the same game component to a game component collection multiple times.

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection.RemoveItem Method

Removes a child object in the collection.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void RemoveItem (  
    int index  
)
```

Parameters

index

The index of the item being removed.

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection.SetItem Method

Modifies the specified child object in the collection.

Overload List

Name	Description
GameComponentCollection.SetItem (Int32, IGameComponent)	Modifies the specified child object in the collection.
GameComponentCollection.SetItem (Int32, IGameComponent)	(Inherited from Collection .)

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollection.SetItem Method (Int32, IGameComponent)

Modifies the specified child object in the collection.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void SetItem (  
    int index,  
    IGameComponent item  
)
```

Parameters

index

The position in the collection.

item

The child object being modified.

Exceptions

Exception type	Condition
NotSupportedException	Cannot set a value using SetItem on GameComponentCollection . Use Add or Remove instead.

See Also

Reference

[GameComponentCollection Class](#)



[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also



Reference

[GameComponentCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollection Events

Public Events

	Name	Description
	ComponentAdded	Raised when a component is added to the GameComponentCollection .
	ComponentRemoved	Raised when a component is removed from the GameComponentCollection .

See Also

Reference

[GameComponentCollection Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollection.ComponentAdded Event

Raised when a component is added to the [GameComponentCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler<GameComponentCollectionEventArgs> ComponentAdded
```

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollection.ComponentRemoved Event

Raised when a component is removed from the [GameComponentCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler<GameComponentCollectionEventArgs> ComponentRemoved
```

See Also

Reference

[GameComponentCollection Class](#)

[GameComponentCollection Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollectionEventArgs Class

Arguments used with events from the [GameComponentCollection](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class GameComponentCollectionEventArgs : EventArgs
```

See Also

Reference

[GameComponentCollectionEventArgs Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


GameComponentCollectionEventArgs Members

The following tables list the members exposed by the GameComponentCollectionEventArgs type.






Public Constructors

	Name	Description
	GameComponentCollectionEventArgs	Creates a new instance of GameComponentCollectionEventArgs.



Public Properties

	Name	Description
	GameComponent	The game component affected by the event.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameComponentCollectionEventArgs Class](#)
[Microsoft.Xna.Framework Namespace](#)

GameComponentCollectionEventArgs Constructor

Creates a new instance of **GameComponentCollectionEventArgs**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameComponentCollectionEventArgs (  
    IGameComponent gameComponent  
)
```

Parameters

gameComponent

The game component affected by the event.

See Also

Reference

[GameComponentCollectionEventArgs Class](#)






[GameComponentCollectionEventArgs Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameComponentCollectionEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[GameComponentCollectionEventArgs Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollectionEventArgs Properties

Public Properties

	Name	Description
	GameComponent	The game component affected by the event.

See Also

Reference

[GameComponentCollectionEventArgs Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameComponentCollectionEventArgs.GameComponent Property

The game component affected by the event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public IGameComponent GameComponent { get; }
```

Property Value

The game component affected by the event.

See Also

Reference

[GameComponentCollectionEventArgs Class](#)

[GameComponentCollectionEventArgs Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameServiceContainer Class

A collection of game services.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class GameServiceContainer : IServiceProvider
```

Remarks

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, [Game.Services](#). Service providers register with [Game.Services](#), and service consumers request services from [Game.Services](#). This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with [Game.Services](#). A service is registered by calling [Game.Services.AddService](#) specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface `IMyService`, you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by [Game.Services.GetService](#) and specifying the desired service. For example, to retrieve [IGraphicsDeviceService](#), you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Services.GetService( t  
typeof(IGraphicsDeviceService) );
```

See Also

Reference

[Game.Services Property](#)

[GameServiceContainer Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









GameServiceContainer Members

The following tables list the members exposed by the GameServiceContainer type.



Public Constructors

Name	Description
 GameServiceContainer	Initializes a new instance of this class, which represents a collection of game services.

Public Methods

Name	Description
 AddService	Adds a service to the GameServiceContainer .
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetService	Gets the object providing a specified service.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 RemoveService	Removes the object providing a specified service.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameServiceContainer Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameServiceContainer Constructor

Initializes a new instance of this class, which represents a collection of game services.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameServiceContainer ()
```

Remarks

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, [Game.Services](#). Service providers register with [Game.Services](#), and service consumers request services from [Game.Services](#). This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with [Game.Services](#). A service is registered by calling [Game.Services.AddService](#) specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface `IMyService`, you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by [Game.Services.GetService](#) and specifying the desired service. For example, to retrieve [IGraphicsDeviceService](#), you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Services.GetService( t  
typeof(IGraphicsDeviceService) );
```

See Also

Reference

[Game.Services Property](#)

[GameServiceContainer Class](#)









[GameServiceContainer Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameServiceContainer Methods

Public Methods

	Name	Description
	AddService	Adds a service to the GameServiceContainer .
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetService	Gets the object providing a specified service.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	RemoveService	Removes the object providing a specified service.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameServiceContainer Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameServiceContainer.AddService Method

Adds a service to the [GameServiceContainer](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void AddService (
    Type type,
    Object provider
)
```

Parameters

type

The type of service to add.

provider

The service provider to add.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>type</i> or <i>provider</i> parameter is null .
ArgumentException	The container already contains a service of this type, or the service provider object of the type requested is not assignable to this service type.

Remarks

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, [Game.Services](#). Service providers register with [Game.Services](#), and service consumers request services from [Game.Services](#). This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with [Game.Services](#). A service is registered by calling [Game.Services.AddService](#) specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface `IMyService`, you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by [Game.Services.GetService](#) and specifying the desired service. For example, to retrieve [IGraphicsDeviceService](#), you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService) Services.GetService( t
ypeof(IGraphicsDeviceService) );
```

See Also

Concepts

[Application Model Overview](#)

Reference

[GameServiceContainer Class](#)

[GameServiceContainer Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameServiceContainer.GetService Method

Gets the object providing a specified service.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public Object GetService (
    Type type
)
```

Parameters

type

The type of service.

Return Value

The object providing the service.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>type</i> parameter is null .

Remarks

Game services are a mechanism for maintaining loose coupling between objects that need to interact with each other. Services work through a mediator—in this case, [Game.Services](#). Service providers register with [Game.Services](#), and service consumers request services from [Game.Services](#). This arrangement allows an object that requires a service to request the service without knowing the name of the service provider.

Game services are defined by an interface. A class specifies the services it provides by implementing interfaces and registering the services with [Game.Services](#). A service is registered by calling [Game.Services.AddService](#) specifying the type of service being implemented and a reference to the object providing the service. For example, to register an object that provides a service represented by the interface `IMyService`, you would use the following code.

```
Services.AddService( typeof( IMyService ), myobject );
```

Once a service is registered, the object providing the service can be retrieved by [Game.Services.GetService](#) and specifying the desired service. For example, to retrieve [IGraphicsDeviceService](#), you would use the following code.

```
IGraphicsDeviceService graphicservice = (IGraphicsDeviceService)Services.GetService( t
ypeof(IGraphicsDeviceService) );
```

See Also

Concepts

[Application Model Overview](#)

Reference

[GameServiceContainer Class](#)

[GameServiceContainer Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameServiceContainer.RemoveService Method

Removes the object providing a specified service.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void RemoveService (  
    Type type  
)
```

Parameters

type

The type of service.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>type</i> parameter is null .

See Also

Reference

[GameServiceContainer Class](#)

[GameServiceContainer Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameTime Class

Snapshot of the game timing state expressed in values that can be used by variable-step (real time) or fixed-step (game time) games.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class GameTime
```

See Also

Tasks

[How To: Make a Game Use a Variable Time Step](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GameTime Members

The following tables list the members exposed by the GameTime type.






Public Constructors

Name	Description
 GameTime	Overloaded. Creates a new instance of GameTime .



Public Properties

Name	Description
 ElapsedGameTime	The amount of elapsed game time since the last update.
 ElapsedRealTime	The amount of elapsed real time (wall clock) since the last frame.
 IsRunningSlowly	Gets a value indicating that the game loop is taking longer than its TargetElapsedTime . In this case, the game loop can be considered to be running too slowly and should do something to "catch up."
 TotalGameTime	The amount of game time since the start of the game.
 TotalRealTime	The amount of real time (wall clock) since the start of the game.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameTime Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameTime Constructor

Creates a new instance of [GameTime](#).

Overload List

Name	Description
GameTime ()	Creates a new instance of GameTime .
GameTime (TimeSpan, TimeSpan, TimeSpan, TimeSpan)	Creates a new instance of GameTime .
GameTime (TimeSpan, TimeSpan, TimeSpan, TimeSpan, Boolean)	Creates a new instance of GameTime .

See Also

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

GameTime Constructor ()

Creates a new instance of **GameTime**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameTime ()
```

See Also

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameTime Constructor (TimeSpan, TimeSpan, TimeSpan, TimeSpan)

Creates a new instance of **GameTime**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameTime (  
    TimeSpan totalRealTime,  
    TimeSpan elapsedRealTime,  
    TimeSpan totalGameTime,  
    TimeSpan elapsedGameTime  
)
```

Parameters

totalRealTime

The amount of real time (wall clock) since the start of the game.

elapsedRealTime

The amount of elapsed real time (wall clock) since the last frame.

totalGameTime

The amount of game time since the start of the game.

elapsedGameTime

The amount of elapsed game time since the last update.

See Also

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameTime Constructor (TimeSpan, TimeSpan, TimeSpan, TimeSpan, Boolean)

Creates a new instance of **GameTime**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GameTime (  
    TimeSpan totalRealTime,  
    TimeSpan elapsedRealTime,  
    TimeSpan totalGameTime,  
    TimeSpan elapsedGameTime,  
    bool isRunningSlowly  
)
```

Parameters

totalRealTime

The amount of real time (wall clock) since the start of the game.

elapsedRealTime

The amount of elapsed real time (wall clock) since the last frame.

totalGameTime

The amount of game time since the start of the game.

elapsedGameTime

The amount of elapsed game time since the last update.

isRunningSlowly

Whether the game is running multiple updates this frame.

See Also

Reference

[GameTime Class](#)






[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameTime Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[GameTime Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameTime Properties

Public Properties

	Name	Description
	ElapsedGameTime	The amount of elapsed game time since the last update.
	ElapsedRealTime	The amount of elapsed real time (wall clock) since the last frame.
	IsRunningSlowly	Gets a value indicating that the game loop is taking longer than its TargetElapsedTime . In this case, the game loop can be considered to be running too slowly and should do something to "catch up."
	TotalGameTime	The amount of game time since the start of the game.
	TotalRealTime	The amount of real time (wall clock) since the start of the game.

See Also

Reference

[GameTime Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameTime.ElapsedGameTime Property

The amount of elapsed game time since the last update.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public TimeSpan ElapsedGameTime { get; set; }
```

Property Value

Elapsed game time since the last update.

RemarksFixed-step clocks update by a fixed time span upon every clock step. This results in uniform clock steps that may not actually track the wall clock time. Fixed-step clocks were popular on console systems where one had tight control over code and a fixed system performance. Fixed-step clocks are also useful when trying to achieve deterministic updates for debugging, offline rendering, or other such scenarios.

See Also

Tasks

[How To: Make a Game Use a Variable Time Step](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GameTime.ElapsedRealTime Property

The amount of elapsed real time (wall clock) since the last frame.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public TimeSpan ElapsedRealTime { get; set; }
```

Property Value

Elapsed real time since the last frame.

RemarksVariable-step clocks automatically track the difference in real-time (wall clock) that transpired since the last clock step.

This type of game timing is popular on PC games where the variable PC configurations affect update time.

See Also

Tasks

[How To: Make a Game Use a Variable Time Step](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GameTime.IsRunningSlowly Property

Gets a value indicating that the game loop is taking longer than its [TargetElapsedTime](#). In this case, the game loop can be considered to be running too slowly and should do something to "catch up."

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool IsRunningSlowly { get; set; }
```

Property Value

true if the game loop is taking too long; **false** otherwise.

See Also

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameTime.TotalGameTime Property

The amount of game time since the start of the game.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public TimeSpan TotalGameTime { get; set; }
```

Property Value

Game time since the start of the game.

RemarksFixed-step clocks update by a fixed time span upon every clock step. This results in uniform clock steps that may not actually track the wall clock time. Fixed step clocks were popular on console systems where one had tight control over code and a fixed system performance. Fixed-step clocks are also useful when trying to achieve deterministic updates for debugging, offline rendering, or other such scenarios.

See Also

Tasks

[How To: Make a Game Use a Variable Time Step](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GameTime.TotalRealTime Property

The amount of real time (wall clock) since the start of the game.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public TimeSpan TotalRealTime { get; set; }
```

Property Value

Real time since the start of the game.

RemarksVariable-step clocks automatically track the difference in real-time (wall clock) that transpired since the last clock step.

This type of game timing is popular on PC games where the variable PC configurations affect update time.

See Also

Tasks

[How To: Make a Game Use a Variable Time Step](#)

Concepts

[Application Model Overview](#)

Reference

[GameTime Class](#)

[GameTime Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow Class

The system window associated with a [Game](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract class GameWindow
```

See Also

Reference

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide






[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GameWindow Members

The following tables list the members exposed by the GameWindow type.









Public Properties

Name	Description
 AllowUserResizing	Specifies whether to allow the user to resize the game window.
 ClientBounds	The screen dimensions of the game window's client rectangle.
 Handle	Gets the handle to the system window.
 ScreenDeviceName	Gets the device name of the screen the window is currently in.
 Title	Gets and sets the title of the system window.



Public Methods

Name	Description
 BeginScreenDeviceChange	Starts a device transition (windowed to full screen or vice versa).
 EndScreenDeviceChange	Overloaded. Completes a device transition.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 OnActivated	Called when the GameWindow gets focus.
 OnClientSizeChanged	Called when the size of the client window changes. Raises the ClientSizeChanged event.
 OnDeactivated	Called when the GameWindow loses focus.
 OnPaint	Called when the GameWindow needs to be painted.
 OnScreenDeviceNameChanged	Called when the GameWindow is moved to a different screen. Raises the ScreenDeviceNameChanged event.
 SetTitle	Sets the title of the GameWindow .

Public Events

Name	Description
 ClientSizeChanged	Raised when the size of the GameWindow changes.
 ScreenDeviceNameChanged	Raised when the GameWindow moves to a different display.

See Also








Reference

[GameWindow Class](#)









[Microsoft.Xna.Framework Namespace](#)

GameWindow Methods

Public Methods

	Name	Description
	BeginScreenDeviceChange	Starts a device transition (windowed to full screen or vice versa).
	EndScreenDeviceChange	Overloaded. Completes a device transition.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	OnActivated	Called when the GameWindow gets focus.
	OnClientSizeChanged	Called when the size of the client window changes. Raises the ClientSizeChanged event.
	OnDeactivated	Called when the GameWindow loses focus.
	OnPaint	Called when the GameWindow needs to be painted.
	OnScreenDeviceNameChanged	Called when the GameWindow is moved to a different screen. Raises the ScreenDeviceNameChanged event.
	SetTitle	Sets the title of the GameWindow .

See Also

Reference

[GameWindow Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameWindow.BeginScreenDeviceChange Method

Starts a device transition (windowed to full screen or vice versa).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract void BeginScreenDeviceChange (  
    bool willBeFullScreen  
)
```

Parameters

willBeFullScreen

Specifies whether the device will be in full-screen mode upon completion of the change.

Remarks This method must be called before any device change occurs (that is, before resetting the graphics device).

[EndScreenDeviceChange](#) should be called to complete the transition.

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.EndScreenDeviceChange Method

Completes a device transition.

Overload List

Name	Description
GameWindow.EndScreenDeviceChange (String)	Completes a device transition.
GameWindow.EndScreenDeviceChange (String, Int32, Int32)	Completes a device transition.

See Also

Reference

[BeginScreenDeviceChange](#)

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

GameWindow.EndScreenDeviceChange Method (String)

Completes a device transition.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void EndScreenDeviceChange (  
    string screenDeviceName  
)
```

Parameters

screenDeviceName

The desktop screen to move the window to. This should be the screen device name of the graphics device that has transitioned to full screen.

RemarksThis method must be called after any device change (that is, after resetting the graphics device).

See Also

Reference

[BeginScreenDeviceChange](#)

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.EndScreenDeviceChange Method (String, Int32, Int32)

Completes a device transition.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract void EndScreenDeviceChange (  
    string screenDeviceName,  
    int clientWidth,  
    int clientHeight  
)
```

Parameters

screenDeviceName

The desktop screen to move the window to. This should be the screen device name of the graphics device that has transitioned to full screen.

clientWidth

The new width of the game's client window.

clientHeight

The new height of the game's client window.

Remarks This method must be called after any device change (that is, after resetting the graphics device).

See Also

Reference

[BeginScreenDeviceChange](#)

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.OnActivated Method

Called when the [GameWindow](#) gets focus.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected void OnActivated ()
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.OnClientSizeChanged Method

Called when the size of the client window changes. Raises the [ClientSizeChanged](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected void OnClientSizeChanged ()
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.OnDeactivated Method

Called when the [GameWindow](#) loses focus.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected void OnDeactivated ()
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.OnPaint Method

Called when the [GameWindow](#) needs to be painted.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected void OnPaint ()
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.OnScreenDeviceNameChanged Method

Called when the [GameWindow](#) is moved to a different screen. Raises the [ScreenDeviceNameChanged](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected void OnScreenDeviceNameChanged ()
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.SetTitle Method

Sets the title of the [GameWindow](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected abstract void SetTitle (  
    string title  
)
```

Parameters

title

The new title of the [GameWindow](#).

Note

This method is ignored on the Xbox 360.

See Also

Reference

[GameWindow Class](#)






[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow Properties

Public Properties

	Name	Description
	AllowUserResizing	Specifies whether to allow the user to resize the game window.
	ClientBounds	The screen dimensions of the game window's client rectangle.
	Handle	Gets the handle to the system window.
	ScreenDeviceName	Gets the device name of the screen the window is currently in.
	Title	Gets and sets the title of the system window.

See Also

Reference

[GameWindow Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameWindow.AllowUserResizing Property

Specifies whether to allow the user to resize the game window.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
[DefaultValueAttribute(false)]  
public abstract bool AllowUserResizing { get; set; }
```

Property Value

true if user resizing is allowed; **false** otherwise.

Note

On the Xbox 360, this property is always **false**.

See Also

Reference

[Game Class](#)

[Game Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Allow the Player to Resize a Game Window](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.ClientBounds Property

The screen dimensions of the game window's client rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract Rectangle ClientBounds { get; }
```

Property Value

Screen dimensions of the game window's client rectangle.

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.Handle Property

Gets the handle to the system window.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract IntPtr Handle { get; }
```

Property Value

The handle to the system window.

Note

On the Xbox 360 this property is always **IntPtr.Zero**.

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.ScreenDeviceName Property

Gets the device name of the screen the window is currently in.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract string ScreenDeviceName { get; }
```

Property Value

The device name of the screen.

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.Title Property

Gets and sets the title of the system window.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public string Title { get; set; }
```

Property Value

The title of the system window.

Exceptions

Exception type	Condition
ArgumentNullException	<i>Title</i> is null . Use an empty string instead.

Note

This property is ignored on the Xbox 360.

See Also

Reference

[GameWindow Class](#)



[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow Events

Public Events

	Name	Description
	ClientSizeChanged	Raised when the size of the GameWindow changes.
	ScreenDeviceNameChanged	Raised when the GameWindow moves to a different display.

See Also

Reference

[GameWindow Class](#)

[Microsoft.Xna.Framework Namespace](#)

GameWindow.ClientSizeChanged Event

Raised when the size of the [GameWindow](#) changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler ClientSizeChanged
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Allow the Player to Resize a Game Window](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameWindow.ScreenDeviceNameChanged Event

Raised when the [GameWindow](#) moves to a different display.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler ScreenDeviceNameChanged
```

See Also

Reference

[GameWindow Class](#)

[GameWindow Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation Class

Holds the settings for creating a graphics device on Windows.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class GraphicsDeviceInformation
```

See Also

Reference

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[3D Graphics Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




GraphicsDeviceInformation Members

The following tables list the members exposed by the GraphicsDeviceInformation type.







Public Constructors

	Name	Description
	GraphicsDeviceInformation	Initializes a new instance of this class.



Public Properties

	Name	Description
	Adapter	Specifies which graphics adapter to create the device on.
	DeviceType	Specifies the type of device to create (hardware, reference, null).
	PresentationParameters	Specifies the presentation parameters to use when creating a graphics device.

Public Methods

	Name	Description
	Clone	Creates a clone of this object.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceInformation Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceInformation Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsDeviceInformation ()
```

See Also

Reference

[GraphicsDeviceInformation Class](#)







[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation Methods

Public Methods

	Name	Description
	Clone	Creates a clone of this object.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceInformation Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceInformation.Clone Method

Creates a clone of this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsDeviceInformation Clone ()
```

Return Value

The cloned object.

See Also

Reference

[GraphicsDeviceInformation Class](#)

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GraphicsDeviceInformation.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GraphicsDeviceInformation.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceInformation Class](#)

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceInformation.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [GraphicsDeviceInformation](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[GraphicsDeviceInformation Class](#)

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GraphicsDeviceInformation Class](#)




[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation Properties

Public Properties

	Name	Description
	Adapter	Specifies which graphics adapter to create the device on.
	DeviceType	Specifies the type of device to create (hardware, reference, null).
	PresentationParameters	Specifies the presentation parameters to use when creating a graphics device.

See Also

Reference

[GraphicsDeviceInformation Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceInformation.Adapter Property

Specifies which graphics adapter to create the device on.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsAdapter Adapter { get; set; }
```

Property Value

The graphics adapter to create the device on.

Exceptions

Exception type	Condition
ArgumentNullException	<i>Adapter</i> is null . Try using DefaultAdapter instead.

See Also

Reference

[GraphicsDeviceInformation Class](#)

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation.DeviceType Property

Specifies the type of device to create (hardware, reference, null).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public DeviceType DeviceType { get; set; }
```

Property Value

The type of device to create.

See Also

Reference

[GraphicsDeviceInformation Class](#)

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceInformation.PresentationParameters Property

Specifies the presentation parameters to use when creating a graphics device.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public PresentationParameters PresentationParameters { get; set; }
```

Property Value

The presentation parameters to use when creating a graphics device.

See Also

Reference

[GraphicsDeviceInformation Class](#)

[GraphicsDeviceInformation Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager Class

Handles the configuration and management of the graphics device.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class GraphicsDeviceManager : IGraphicsDeviceService, IDisposable, IGraphicsDeviceMa  
nager
```

Remarks

Custom behavior of the [GraphicsDeviceManager](#) can be achieved by deriving a class from [GraphicsDeviceManager](#). For example, to allow only widescreen devices in full-screen mode the [RankDevices](#) method could be overridden to drop non-widescreen devices (see [How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)).

Or, if you are creating a 2D game, you might want to create a graphics device without a depth buffer. By adding an event handler for the [PreparingDeviceSettings](#) event to the [GraphicsDeviceManager](#), you can control the [PresentationParameters](#) properties associated with the graphics device. In the example below, the event handler is named **graphics_Settings_NoDepthBuffer**.

C#

```
graphics = new GraphicsDeviceManager(this);  
graphics.PreparingDeviceSettings +=  
    new EventHandler<PreparingDeviceSettingsEventArgs>(graphics_Settings_NoDepthBuffer);
```

To ensure that no depth buffer is created, your event handler can set the [EnableAutoDepthStencil](#) property of the graphic device's [PresentationParameters](#) to **false**, as shown below.

C#

```
void graphics_Settings_NoDepthBuffer(object sender,  
    PreparingDeviceSettingsEventArgs e)  
{  
    e.GraphicsDeviceInformation.PresentationParameters.  
        EnableAutoDepthStencil = false;  
}
```

See Also

Reference

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Application Model Overview](#)

[3D Graphics Overview](#)

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)


[How To: Display a Game in Full-Screen Mode](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GraphicsDeviceManager Members

The following tables list the members exposed by the GraphicsDeviceManager type.











Public Constructors

Name	Description
 GraphicsDeviceManager	Creates a new GraphicsDeviceManager and registers it to handle the configuration and management of the graphics device for the specified Game .








Public Fields

Name	Description
 DefaultBackBufferHeight	Specifies the default minimum back-buffer height.
 DefaultBackBufferWidth	Specifies the default minimum back-buffer width.
 ValidAdapterFormats	Specifies the set of adapter formats supported by the GraphicsDeviceManager .
 ValidBackBufferFormats	Specifies the set of back-buffer formats supported by the GraphicsDeviceManager .
 ValidDeviceTypes	Specifies the set of valid device types supported by the GraphicsDeviceManager .









Public Properties




Name	Description
 GraphicsDevice	Gets the GraphicsDevice associated with the GraphicsDeviceManager .
 IsFullScreen	Gets or sets a value that indicates whether the device should start in full-screen mode.
 MinimumPixelShaderProfile	Gets or sets the minimum pixel shader version required by the GraphicsDeviceManager .
 MinimumVertexShaderProfile	Gets or sets the minimum vertex shader version required by the GraphicsDeviceManager .
 PreferMultiSampling	Gets or sets a value that indicates whether to enable a multisampled back buffer.
 PreferredBackBufferFormat	Gets or sets the format of the back buffer.
 PreferredBackBufferHeight	Gets or sets the preferred back-buffer height.
 PreferredBackBufferWidth	Gets or sets the preferred back-buffer width.
 PreferredDepthStencilFormat	Gets or sets the format of the depth stencil.
 SynchronizeWithVerticalTrace	Gets or sets a value that indicates whether to sync to the vertical trace (vsync) when presenting the back buffer.

Public Methods







Name	Description
 ApplyChanges	Applies any changes to device-related properties, changing the graphics device as necessary.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToggleFullScreen	Toggles between full screen and windowed mode.
 ToString	(Inherited from Object .)

Protected Methods





Name	Description
 CanResetDevice	Determines whether the given GraphicsDeviceInformation is compatible with the existing graphics device.
 Dispose	Releases the unmanaged resources used by the GraphicsDeviceManager and optionally releases the managed resources.
 Finalize	(Inherited from Object .)
 FindBestDevice	Finds the best device configuration that is compatible with the current device preferences.
 MemberwiseClone	(Inherited from Object .)
 OnDeviceCreated	Called when a device is created. Raises the DeviceCreated event.
 OnDeviceDisposing	Called when a device is being disposed. Raises the DeviceDisposing event.
 OnDeviceReset	Called when the device has been reset. Raises the DeviceReset event.

 OnDeviceResetting	Called when the device is about to be reset. Raises the DeviceResetting event.
 OnPreparingDeviceSettings	Called when the GraphicsDeviceManager is changing the GraphicsDevice settings (during reset or recreation of the GraphicsDevice). Raises the PreparingDeviceSettings event.
 RankDevices	Ranks the given list of devices that satisfy the given preferences.

Public Events

Name	Description
 DeviceCreated	Raised when a new graphics device is created.
 DeviceDisposing	Raised when the GraphicsDeviceManager is being disposed.
 DeviceReset	Raised when the GraphicsDeviceManager is reset.
 DeviceResetting	Raised when the GraphicsDeviceManager is about to be reset.
 Disposed	Raised when the GraphicsDeviceManager is disposed.
 PreparingDeviceSettings	Raised when the GraphicsDeviceManager is changing the GraphicsDevice settings (during reset or recreation of the GraphicsDevice).

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.GraphicsDeviceManager.BeginDraw	Prepares the GraphicsDevice to draw.
 Microsoft.Xna.Framework.GraphicsDeviceManager.CreateDevice	Called to ensure that the device manager has created a valid device.
 System.IDisposable.Dispose	Releases all resources used by the GraphicsDeviceManager class.
 Microsoft.Xna.Framework.GraphicsDeviceManager.EndDraw	Called by the game at the end of drawing and presents the final rendering.

See Also











Reference

[GraphicsDeviceManager Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceManager Fields

Public Fields

	Name	Description
 	DefaultBackBufferHeight	Specifies the default minimum back-buffer height.
 	DefaultBackBufferWidth	Specifies the default minimum back-buffer width.
 	ValidAdapterFormats	Specifies the set of adapter formats supported by the GraphicsDeviceManager .
 	ValidBackBufferFormats	Specifies the set of back-buffer formats supported by the GraphicsDeviceManager .
 	ValidDeviceTypes	Specifies the set of valid device types supported by the GraphicsDeviceManager .

See Also

Reference

[GraphicsDeviceManager Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceManager.DefaultBackBufferHeight Field

Specifies the default minimum back-buffer height.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public static readonly int DefaultBackBufferHeight
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.DefaultBackBufferWidth Field

Specifies the default minimum back-buffer width.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public static readonly int DefaultBackBufferWidth
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.ValidAdapterFormats Field

Specifies the set of adapter formats supported by the [GraphicsDeviceManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public static readonly SurfaceFormat[] ValidAdapterFormats
```

RemarksThe order of the list determines the preference of any given format.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.ValidBackBufferFormats Field

Specifies the set of back-buffer formats supported by the [GraphicsDeviceManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public static readonly SurfaceFormat[] ValidBackBufferFormats
```

RemarksThe order of the list determines the preference of any given format.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.ValidDeviceTypes Field

Specifies the set of valid device types supported by the [GraphicsDeviceManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public static readonly DeviceType[] ValidDeviceTypes
```

RemarksThe order of the list determines the preference of any given device type.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager Constructor

Creates a new [GraphicsDeviceManager](#) and registers it to handle the configuration and management of the graphics device for the specified [Game](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsDeviceManager (
    Game game
)
```

Parameters

game

[Game](#) the [GraphicsDeviceManager](#) should be associated with.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>game</i> parameter is null .
ArgumentException	A graphics device manager has already been added to Services and registered as the graphics device manager for the application. The graphics device manager cannot be changed once it is set.

Remarks

This constructor adds this [GraphicsDeviceManager](#) to the collection of game services that have been implicitly registered by adding them to the [Game.Services](#) property.

⚠ Caution

Although this constructor requires an instance of [Game](#), if the application does not otherwise require an instance of [Game](#) it is often better to implement a the [IGraphicsDeviceService](#) and [IGraphicsDeviceManager](#) interfaces to provide the same services that would be provided by the [GraphicsDeviceManager](#).

Custom behavior of the [GraphicsDeviceManager](#) can be achieved by deriving a class from [GraphicsDeviceManager](#). For example, to allow only widescreen devices in full-screen mode the [RankDevices](#) method could be overridden to drop non-widescreen devices (see [How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)).

Or, if you are creating a 2D game, you might want to create a graphics device without a depth buffer. By adding an event handler for the [PreparingDeviceSettings](#) event to the [GraphicsDeviceManager](#), you can control the [PresentationParameters](#) properties associated with the graphics device. In the example below, the event handler is named **graphics_Settings_NoDepthBuffer**.

C#

```
graphics = new GraphicsDeviceManager(this);
graphics.PreparingDeviceSettings +=
    new EventHandler<PreparingDeviceSettingsEventArgs>(
        graphics_Settings_NoDepthBuffer);
```

To ensure that no depth buffer is created, your event handler can set the [EnableAutoDepthStencil](#) property of the graphic device's [PresentationParameters](#) to **false**, as shown below.

C#

```
void graphics_Settings_NoDepthBuffer(object sender,
    PreparingDeviceSettingsEventArgs e)
{
    e.GraphicsDeviceInformation.PresentationParameters.
        EnableAutoDepthStencil = false;
}
```

See Also

Reference

[GraphicsDeviceManager Class](#)








[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)












Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager Methods





Public Methods

Name	Description
 ApplyChanges	Applies any changes to device-related properties, changing the graphics device as necessary.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToggleFullScreen	Toggles between full screen and windowed mode.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 CanResetDevice	Determines whether the given GraphicsDeviceInformation is compatible with the existing graphics device.
 Dispose	Releases the unmanaged resources used by the GraphicsDeviceManager and optionally releases the managed resources.
 Finalize	(Inherited from Object .)
 FindBestDevice	Finds the best device configuration that is compatible with the current device preferences.
 MemberwiseClone	(Inherited from Object .)
 OnDeviceCreated	Called when a device is created. Raises the DeviceCreated event.
 OnDeviceDisposing	Called when a device is being disposed. Raises the DeviceDisposing event.
 OnDeviceReset	Called when the device has been reset. Raises the DeviceReset event.
 OnDeviceResetting	Called when the device is about to be reset. Raises the DeviceResetting event.
 OnPreparingDeviceSettings	Called when the GraphicsDeviceManager is changing the GraphicsDevice settings (during reset or recreation of the GraphicsDevice). Raises the PreparingDeviceSettings event.
 RankDevices	Ranks the given list of devices that satisfy the given preferences.

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.IGraphicsDeviceManager.BeginDraw	Prepares the GraphicsDevice to draw.
 Microsoft.Xna.Framework.IGraphicsDeviceManager.CreateDevice	Called to ensure that the device manager has created a valid device.
 System.IDisposable.Dispose	Releases all resources used by the GraphicsDeviceManager class.
 Microsoft.Xna.Framework.IGraphicsDeviceManager.EndDraw	Called by the game at the end of drawing and presents the final rendering.

See Also

Reference

[GraphicsDeviceManager Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceManager.ApplyChanges Method

Applies any changes to device-related properties, changing the graphics device as necessary.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void ApplyChanges ()
```

Exceptions

Exception type	Condition
NoSuitableGraphicsDeviceException	<p>One of the following conditions has occurred:</p> <ul style="list-style-type: none">• Could not find a Direct3D device that has a Direct3D9-level driver and supports a minimum of Pixel Shader Model 1.1.• Could not find a Direct3D device compatible with the current device preferences.• The process of ranking devices removed all compatible devices.• Direct3D hardware acceleration is not available or has been disabled. Verify that a Direct3D-enabled graphics device is installed, and check the display properties to verify that Hardware acceleration is set to Full.• Remote Desktop is in use. Direct3D is not available when you are using Remote Desktop.
InvalidOperationException	<p>This graphics device manager has not been added to Services and registered as the graphics device manager for the application. Device settings cannot be changed or committed unless the device has been registered.</p>

<p>ArgumentException</p>	<p>Invalid arguments have been requested for the creation of the graphics device.</p> <p>Invalid back buffer options have been requested:</p> <ul style="list-style-type: none"> • BackBufferCount is not between 0 and 3. • BackBufferFormat is not valid. Try SurfaceFormat.Bgr565, SurfaceFormat.Bgr555, SurfaceFormat.Bgra5551, SurfaceFormat.Bgr32, SurfaceFormat.Color, SurfaceFormat.Bgra1010102. • The selected BackBufferFormat and IsFullScreen values are not valid for the selected adapter format and device type. • The BackBufferWidth and BackBufferHeight values do not correspond to a valid display mode. • FullScreenRefreshRateInHz does not match a valid display mode for the specified BackBufferWidth and BackBufferHeight. <p>Invalid PresentationParameters.PresentationInterval or SwapEffect options have been requested:</p> <ul style="list-style-type: none"> • On Xbox 360, PresentationParameters.PresentationInterval has been assigned the invalid value PresentInterval.Four. • SwapEffect.Copy has been assigned to the PresentationParameters.SwapEffect property, but BackBufferCount is not 1. When using SwapEffect.Copy, BackBufferCount must be 1. • SwapEffect is not one of the following: SwapEffect.Copy, SwapEffect.Discard, or SwapEffect.Flip. <p>Invalid multisampling options have been requested:</p> <ul style="list-style-type: none"> • Multisampling has been enabled, but SwapEffect.Discard has not been specified for the SwapEffect property. SwapEffect.Discard is required when enabling multisampling. • The selected MultiSampleType is not compatible with the current BackBufferFormat and the IsFullScreen value for the selected adapter. • The selected MultiSampleQuality value is invalid for the selected MultiSampleType. <p>Invalid depth stencil options have been requested:</p> <ul style="list-style-type: none"> • AutoDepthStencilFormat is not supported as a depth/stencil format when using the selected adapter BackBufferFormat. • PresentOptions has been set to PresentOptions.DiscardDepthStencil, but EnableAutoDepthStencil is not true. • Multiple PresentationParameters have been specified and EnableAutoDepthStencil is true, but EnableAutoDepthStencil, BackBufferFormat, BackBufferWidth, and BackBufferHeight are not the same for all adapters. <p>IsFullScreen is true, but:</p> <ul style="list-style-type: none"> • BackBufferWidth and BackBufferHeight are not nonzero. • PresentationInterval is not one of the following: PresentInterval.Default, PresentInterval.Immediate, PresentInterval.One, PresentInterval.Two, PresentInterval.Three, or PresentInterval.Four. • FullScreenRefreshRateInHz is not nonzero. • The selected adapter does not support the selected full screen PresentationInterval. <p>IsFullScreen is false, but:</p> <ul style="list-style-type: none"> • PresentationInterval is not one of the following: PresentInterval.Default, PresentInterval.Immediate, or PresentInterval.One. • FullScreenRefreshRateInHz is not zero.
--------------------------	--

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.CanResetDevice Method

Determines whether the given [GraphicsDeviceInformation](#) is compatible with the existing graphics device.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual bool CanResetDevice (  
    GraphicsDeviceInformation newDeviceInfo  
)
```

Parameters

newDeviceInfo

Information describing the desired device configuration.

Return Value

true if a device reset can be used to get a device compatible with the [GraphicsDeviceInformation](#); **false** otherwise.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.Dispose Method

Releases the unmanaged resources used by the [GraphicsDeviceManager](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both automatic and manual resources; **false** to release only manual resources.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.FindBestDevice Method

Finds the best device configuration that is compatible with the current device preferences.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual GraphicsDeviceInformation FindBestDevice (  
    bool anySuitableDevice  
)
```

Parameters

anySuitableDevice

true if the [FindBestDevice](#) can select devices from any available adapter; **false** if only the current adapter should be considered.

Return Value

The best device configuration found.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.IGraphicsDeviceManager.BeginDraw Method

Prepares the [GraphicsDevice](#) to draw.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
private bool Microsoft.Xna.Framework.IGraphicsDeviceManager.BeginDraw ()
```

Return Value

true if the [GraphicsDevice](#) is ready to draw; **false** otherwise.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.IGraphicsDeviceManager.CreateDevice Method

Called to ensure that the device manager has created a valid device.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.IGraphicsDeviceManager.CreateDevice ()
```

Exceptions

Exception type	Condition
NoSuitableGraphicsDeviceException	<p>One of the following conditions has occurred:</p> <ul style="list-style-type: none"> • Could not find a Direct3D device that has a Direct3D9-level driver and supports a minimum of Pixel Shader Model 1.1. • Could not find a Direct3D device compatible with the current device preferences. • The process of ranking devices removed all compatible devices. • Direct3D hardware acceleration is not available or has been disabled. Verify that a Direct3D-enabled graphics device is installed, and check the display properties to verify that Hardware acceleration is set to Full. • Remote Desktop is in use. Direct3D is not available when you are using Remote Desktop.
InvalidOperationException	<p>This graphics device manager has not been added to Services and registered as the graphics device manager for the application. Device settings cannot be changed or committed unless the device has been registered.</p>

ArgumentException

Invalid arguments have been requested for the creation of the graphics device.

Invalid back buffer options have been requested:

- [BackBufferCount](#) is not between 0 and 3.
- [BackBufferFormat](#) is not valid. Try [SurfaceFormat.Bgr565](#), [SurfaceFormat.Bgr555](#), [SurfaceFormat.Bgra5551](#), [SurfaceFormat.Bgr32](#), [SurfaceFormat.Color](#), [SurfaceFormat.Bgra1010102](#).
- The selected [BackBufferFormat](#) and [IsFullScreen](#) values are not valid for the selected adapter format and device type.
- The [BackBufferWidth](#) and [BackBufferHeight](#) values do not correspond to a valid display mode.
- [FullScreenRefreshRateInHz](#) does not match a valid display mode for the specified [BackBufferWidth](#) and [BackBufferHeight](#).

Invalid [PresentationParameters.PresentationInterval](#) or [SwapEffect](#) options have been requested:

- On Xbox 360, [PresentationParameters.PresentationInterval](#) has been assigned the invalid value [PresentationInterval.Four](#).
- [SwapEffect.Copy](#) has been assigned to the [PresentationParameters.SwapEffect](#) property, but [BackBufferCount](#) is not 1. When using [SwapEffect.Copy](#), [BackBufferCount](#) must be 1.
- [SwapEffect](#) is not one of the following: [SwapEffect.Copy](#), [SwapEffect.Discard](#), or [SwapEffect.Flip](#).

Invalid multisampling options have been requested:

- Multisampling has been enabled, but [SwapEffect.Discard](#) has not been specified for the [SwapEffect](#) property. [SwapEffect.Discard](#) is required when enabling multisampling.
- The selected [MultiSampleType](#) is not compatible with the current [BackBufferFormat](#) and the [IsFullScreen](#) value for the selected adapter.
- The selected [MultiSampleQuality](#) value is invalid for the selected [MultiSampleType](#).

Invalid depth stencil options have been requested:

- [AutoDepthStencilFormat](#) is not supported as a depth/stencil format when using the selected adapter [BackBufferFormat](#).
- [PresentOptions](#) has been set to [PresentOptions.DiscardDepthStencil](#), but [EnableAutoDepthStencil](#) is not **true**.
- Multiple [PresentationParameters](#) have been specified and [EnableAutoDepthStencil](#) is **true**, but [EnableAutoDepthStencil](#), [BackBufferFormat](#), [BackBufferWidth](#), and [BackBufferHeight](#) are not the same for all adapters.

[IsFullScreen](#) is **true**, but:

- [BackBufferWidth](#) and [BackBufferHeight](#) are not nonzero.
- [PresentationInterval](#) is not one of the following: [PresentationInterval.Default](#), [PresentationInterval.Immediate](#), [PresentationInterval.One](#), [PresentationInterval.Two](#), [PresentationInterval.Three](#), or [PresentationInterval.Four](#).
- [FullScreenRefreshRateInHz](#) is not nonzero.
- The selected adapter does not support the selected full screen [PresentationInterval](#).

[IsFullScreen](#) is **false**, but:

- [PresentationInterval](#) is not one of the following: [PresentationInterval.Default](#), [PresentationInterval.Immediate](#), or [PresentationInterval.One](#).
- [FullScreenRefreshRateInHz](#) is not zero.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.IGraphicsDeviceManager.EndDraw Method

Called by the game at the end of drawing and presents the final rendering.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.IGraphicsDeviceManager.EndDraw ()
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.OnDeviceCreated Method

Called when a device is created. Raises the [DeviceCreated](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnDeviceCreated (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [GraphicsDeviceManager](#).

args

Arguments for the [DeviceCreated](#) event.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.OnDeviceDisposing Method

Called when a device is being disposed. Raises the [DeviceDisposing](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnDeviceDisposing (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [GraphicsDeviceManager](#).

args

Arguments for the [DeviceDisposing](#) event.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.OnDeviceReset Method

Called when the device has been reset. Raises the [DeviceReset](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnDeviceReset (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [GraphicsDeviceManager](#).

args

Arguments for the [DeviceReset](#) event.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.OnDeviceResetting Method

Called when the device is about to be reset. Raises the [DeviceResetting](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnDeviceResetting (  
    Object sender,  
    EventArgs args  
)
```

Parameters

sender

The [GraphicsDeviceManager](#).

args

Arguments for the [DeviceResetting](#) event.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.OnPreparingDeviceSettings Method

Called when the [GraphicsDeviceManager](#) is changing the [GraphicsDevice](#) settings (during reset or recreation of the [GraphicsDevice](#)). Raises the [PreparingDeviceSettings](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void OnPreparingDeviceSettings (  
    Object sender,  
    PreparingDeviceSettingsEventArgs args  
)
```

Parameters

sender

The [GraphicsDeviceManager](#).

args

The graphics device information to modify.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.RankDevices Method

Ranks the given list of devices that satisfy the given preferences.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
protected virtual void RankDevices (  
    List<GraphicsDeviceInformation> foundDevices  
)
```

Parameters

foundDevices

The list of devices to rank.

Remarks **RankDevices** orders the provided list so that devices earlier in the list are preferred over devices later in the list. This method can remove devices from the list if they do not satisfy some custom criteria.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.IDisposable.Dispose Method

Releases all resources used by the [GraphicsDeviceManager](#) class.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
private void System.IDisposable.Dispose ()
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.ToggleFullScreen Method

Toggles between full screen and windowed mode.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void ToggleFullScreen ()
```

Exceptions

Exception type	Condition
NoSuitableGraphicsDeviceException	<p>One of the following conditions has occurred:</p> <ul style="list-style-type: none"> • Could not find a Direct3D device that has a Direct3D9-level driver and supports a minimum of Pixel Shader Model 1.1. • Could not find a Direct3D device compatible with the current device preferences. • The process of ranking devices removed all compatible devices. • Direct3D hardware acceleration is not available or has been disabled. Verify that a Direct3D-enabled graphics device is installed, and check the display properties to verify that Hardware acceleration is set to Full. • Remote Desktop is in use. Direct3D is not available when you are using Remote Desktop.
InvalidOperationException	<p>This graphics device manager has not been added to Services and registered as the graphics device manager for the application. Device settings cannot be changed or committed unless the device has been registered.</p>
ArgumentException	<p>Full-screen mode is not compatible with the current back buffer settings:</p> <ul style="list-style-type: none"> • The selected BackBufferFormat and IsFullScreen values are not valid for the selected adapter format and device type. • FullScreenRefreshRateInHz does not match a valid display mode for the specified BackBufferWidth and BackBufferHeight. • The selected MultiSampleType is not compatible with the current BackBufferFormat and the IsFullScreen value for the selected adapter. <p>IsFullScreen is true, but:</p> <ul style="list-style-type: none"> • BackBufferWidth and BackBufferHeight are not nonzero. • PresentationInterval is not one of the following: PresentInterval.Default, PresentInterval.Immediate, PresentInterval.One, PresentInterval.Two, PresentInterval.Three, or PresentInterval.Four. • FullScreenRefreshRateInHz is not nonzero. • The selected adapter does not support the selected full screen PresentationInterval. <p>IsFullScreen is false, but:</p> <ul style="list-style-type: none"> • PresentationInterval is not one of the following: PresentInterval.Default, PresentInterval.Immediate, or PresentInterval.One. • FullScreenRefreshRateInHz is not zero.

RemarksThis method has no effect on the Xbox 360.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)











Programming Guide

[How To: Display a Game in Full-Screen Mode](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the GraphicsDevice associated with the GraphicsDeviceManager .
	IsFullScreen	Gets or sets a value that indicates whether the device should start in full-screen mode.
	MinimumPixelShaderProfile	Gets or sets the minimum pixel shader version required by the GraphicsDeviceManager .
	MinimumVertexShaderProfile	Gets or sets the minimum vertex shader version required by the GraphicsDeviceManager .
	PreferMultiSampling	Gets or sets a value that indicates whether to enable a multisampled back buffer.
	PreferredBackBufferFormat	Gets or sets the format of the back buffer.
	PreferredBackBufferHeight	Gets or sets the preferred back-buffer height.
	PreferredBackBufferWidth	Gets or sets the preferred back-buffer width.
	PreferredDepthStencilFormat	Gets or sets the format of the depth stencil.
	SynchronizeWithVerticalTrace	Gets or sets a value that indicates whether to sync to the vertical trace (vsync) when presenting the back buffer.

See Also

Reference

[GraphicsDeviceManager Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceManager.GraphicsDevice Property

Gets the [GraphicsDevice](#) associated with the [GraphicsDeviceManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The [GraphicsDevice](#) associated with the [GraphicsDeviceManager](#).

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.IsFullScreen Property

Gets or sets a value that indicates whether the device should start in full-screen mode.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool IsFullScreen { get; set; }
```

Property Value

Value that indicates whether the device should start in full-screen mode.

Remarks If no devices are present that support full screen, or the underlying host supports only full screen, this value may be ignored. This value has no effect on Xbox 360.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.MinimumPixelShaderProfile Property

Gets or sets the minimum pixel shader version required by the [GraphicsDeviceManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public ShaderProfile MinimumPixelShaderProfile { get; set; }
```

Property Value

The minimum pixel shader version required by the [GraphicsDeviceManager](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p><i>MinimumPixelShaderProfile</i> is not a valid pixel shader profile. Use one of the following:</p> <ul style="list-style-type: none">• ShaderProfile.PS_1_1• ShaderProfile.PS_1_2• ShaderProfile.PS_1_3• ShaderProfile.PS_1_4• ShaderProfile.PS_2_0• ShaderProfile.PS_2_A• ShaderProfile.PS_2_B• ShaderProfile.PS_2_SW• ShaderProfile.PS_3_0• ShaderProfile.XPS_3_0

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.MinimumVertexShaderProfile Property

Gets or sets the minimum vertex shader version required by the [GraphicsDeviceManager](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public ShaderProfile MinimumVertexShaderProfile { get; set; }
```

Property Value

The minimum vertex shader version required by the [GraphicsDeviceManager](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p><i>MinimumVertexShaderProfile</i> is not a valid vertex shader profile. Use one of the following:</p> <ul style="list-style-type: none">• ShaderProfile.VS_1_1• ShaderProfile.VS_2_0• ShaderProfile.VS_2_A• ShaderProfile.VS_2_SW• ShaderProfile.VS_3_0• ShaderProfile.XVS_3_0

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.PreferMultiSampling Property

Gets or sets a value that indicates whether to enable a multisampled back buffer. For more information about multisampling, see [MultiSampleType Enumeration](#) and [What Is Antialiasing?](#)

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool PreferMultiSampling { get; set; }
```

Property Value

Value indicating whether multisampling is enabled on the back buffer.

Remarks **PreferMultiSampling** is ignored if no hardware is available that supports multisampling.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.PreferredBackBufferFormat Property

Gets or sets the format of the back buffer.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public SurfaceFormat PreferredBackBufferFormat { get; set; }
```

Property Value

The format of the back buffer.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The <i>BackBufferFormat</i> requested is not valid. Use one of the following: <ul style="list-style-type: none">• SurfaceFormat.Bgr565• SurfaceFormat.Bgr555• SurfaceFormat.Bgra5551• SurfaceFormat.Bgr32• SurfaceFormat.Color• SurfaceFormat.Bgra1010102

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.PreferredBackBufferHeight Property

Gets or sets the preferred back-buffer height.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public int PreferredBackBufferHeight { get; set; }
```

Property Value

The preferred back-buffer height.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>BackBufferHeight</i> is not greater than zero.

Remarks

If you request a back-buffer resolution that is not supported by the output device, the XNA framework automatically selects the highest resolution supported by the output device. For example, if a graphics back-buffer, with a resolution of 1920x1080 (e.g. 1080p or 1080i), is created and displayed on a device with 480i resolution, the back-buffer is automatically resized to 480i.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Display a Game in Full-Screen Mode](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.PreferredBackBufferWidth Property

Gets or sets the preferred back-buffer width.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public int PreferredBackBufferWidth { get; set; }
```

Property Value

The preferred back-buffer width.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>BackBufferWidth</i> is not greater than zero.

Remarks

If you request a back-buffer resolution that is not supported by the output device, the XNA framework automatically selects the highest resolution supported by the output device. For example, if a graphics back-buffer, with a resolution of 1920x1080 (e.g. 1080p or 1080i), is created and displayed on a device with 480i resolution, the back-buffer is automatically resized to 480i.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Display a Game in Full-Screen Mode](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.PreferredDepthStencilFormat Property

Gets or sets the format of the depth stencil.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public DepthFormat PreferredDepthStencilFormat { get; set; }
```

Property Value

The format of the depth stencil.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The <i>DepthStencilFormat</i> requested is an invalid format. Use one of the following: <ul style="list-style-type: none">• DepthFormat.Depth32• DepthFormat.Depth15Stencil1• DepthFormat.Depth24Stencil8• DepthFormat.Depth24• DepthFormat.Depth24Stencil4• DepthFormat.Depth16• DepthFormat.Depth24Stencil8Single

Remarks

For an example of using this property, see [How To: Draw a Shadow](#) and [What Is a Stencil Buffer?](#)

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.SynchronizeWithVerticalRetrace Property

Gets or sets a value that indicates whether to sync to the vertical trace (vsync) when presenting the back buffer.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool SynchronizeWithVerticalRetrace { get; set; }
```

Property Value

Value that indicates whether to sync to the vertical trace (vsync) when presenting the back buffer.

Remarks This value is honored (when possible) in full screen and windowed modes.

See Also

Reference

[GraphicsDeviceManager Class](#)







[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager Events

Public Events

	Name	Description
	DeviceCreated	Raised when a new graphics device is created.
	DeviceDisposing	Raised when the GraphicsDeviceManager is being disposed.
	DeviceReset	Raised when the GraphicsDeviceManager is reset.
	DeviceResetting	Raised when the GraphicsDeviceManager is about to be reset.
	Disposed	Raised when the GraphicsDeviceManager is disposed.
	PreparingDeviceSettings	Raised when the GraphicsDeviceManager is changing the GraphicsDevice settings (during reset or recreation of the GraphicsDevice).

See Also

Reference

[GraphicsDeviceManager Class](#)

[Microsoft.Xna.Framework Namespace](#)

GraphicsDeviceManager.DeviceCreated Event

Raised when a new graphics device is created.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler DeviceCreated
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.DeviceDisposing Event

Raised when the [GraphicsDeviceManager](#) is being disposed.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler DeviceDisposing
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.DeviceReset Event

Raised when the [GraphicsDeviceManager](#) is reset.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler DeviceReset
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.DeviceResetting Event

Raised when the [GraphicsDeviceManager](#) is about to be reset.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler DeviceResetting
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.Disposed Event

Raised when the [GraphicsDeviceManager](#) is disposed.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public event EventHandler Disposed
```

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceManager.PreparingDeviceSettings Event

Raised when the [GraphicsDeviceManager](#) is changing the [GraphicsDevice](#) settings (during reset or recreation of the [GraphicsDevice](#)).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

```
C#
public event EventHandler<PreparingDeviceSettingsEventArgs> PreparingDeviceSettings
```

Example

Handle **PreparingDeviceSettings** events to set graphics options for your device - such as antialiasing.

C#

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
    graphics.PreferMultiSampling = true;
    graphics.PreparingDeviceSettings +=
        new EventHandler<PreparingDeviceSettingsEventArgs>(
            graphics_PreparingDeviceSettings);
}
```

C#

```
void graphics_PreparingDeviceSettings(object sender,
    PreparingDeviceSettingsEventArgs e)
{
    // Xbox 360 and most PCs support FourSamples/0
    // (4x) and TwoSamples/0 (2x) antialiasing.
    PresentationParameters pp =
        e.GraphicsDeviceInformation.PresentationParameters;

    #if XBOX
        pp.MultiSampleQuality = 0;
        pp.MultiSampleType = MultiSampleType.FourSamples;
        return;
    #else
        int quality = 0;
        GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
        SurfaceFormat format = adapter.CurrentDisplayMode.Format;
        // Check for 4xAA
        if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
            false, MultiSampleType.FourSamples, out quality))
        {
            // even if a greater quality is returned, we only want quality 0
            pp.MultiSampleQuality = 0;
            pp.MultiSampleType =
                MultiSampleType.FourSamples;
        }
        // Check for 2xAA
        else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
            format, false, MultiSampleType.TwoSamples, out quality))
        {
            // even if a greater quality is returned, we only want quality 0
            pp.MultiSampleQuality = 0;
            pp.MultiSampleType =
                MultiSampleType.TwoSamples;
        }
        return;
    #endif
}
```

}

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IDrawable Interface

Defines the interface for a drawable game component.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public interface IDrawable
```

See Also

Reference

[IDrawable Members](#)



[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


IDrawable Members

The following tables list the members exposed by the IDrawable type.



Public Properties

	Name	Description
	DrawOrder	The order in which to draw this object relative to other objects. Objects with a lower value are drawn first.
	Visible	Indicates whether IDrawable.Draw should be called in Game.Draw for this game component.

Public Methods

	Name	Description
	Draw	Draws the IDrawable .

Public Events

	Name	Description
	DrawOrderChanged	Raised when the DrawOrder property changes.
	VisibleChanged	Raised when the Visible property changes.

See Also


Reference

[IDrawable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IDrawable Methods

Public Methods

	Name	Description
	Draw	Draws the IDrawable .

See Also

Reference

[IDrawable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IDrawable.Draw Method

Draws the [IDrawable](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Draw (  
    gameTime  
)
```

Parameters

gameTime

Snapshot of the game's timing state.

Remarks

⚠Caution

In the **Draw** method of an Xbox 360 game, vertex buffers, index buffers, and textures should not be written using **SetData** when they are intended to be used for rendering. This condition may lead to graphics corruption or crashes. To avoid this potential issue, use [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#) as the preferred alternative to [VertexBuffer.SetData](#) for dynamic vertex generation.

There is a good reason why you should use [DrawUserPrimitives](#) or [DrawUserIndexedPrimitives](#). In cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is used on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the **Draw** function are not submitted until [Present](#) is called. (Note that **Draw** implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[IDrawable Interface](#)



[IDrawable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IDrawable Properties

Public Properties

	Name	Description
	DrawOrder	The order in which to draw this object relative to other objects. Objects with a lower value are drawn first.
	Visible	Indicates whether IDrawable.Draw should be called in Game.Draw for this game component.

See Also

Reference

[IDrawable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IDrawable.DrawOrder Property

The order in which to draw this object relative to other objects. Objects with a lower value are drawn first.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract int DrawOrder { get; }
```

Property Value

Order in which to draw this object relative to other objects.

See Also

Reference

[IDrawable Interface](#)

[IDrawable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IDrawable.Visible Property

Indicates whether [IDrawable.Draw](#) should be called in [Game.Draw](#) for this game component.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract bool Visible { get; }
```

Property Value

true if [Draw](#) should be called; **false** otherwise.

See Also

Reference

[IDrawable Interface](#)



[IDrawable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IDrawable Events

Public Events

	Name	Description
	DrawOrderChanged	Raised when the DrawOrder property changes.
	VisibleChanged	Raised when the Visible property changes.

See Also

Reference

[IDrawable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IDrawable.DrawOrderChanged Event

Raised when the [DrawOrder](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract event EventHandler DrawOrderChanged
```

See Also

Reference

[IDrawable Interface](#)

[IDrawable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IDrawable.VisibleChanged Event

Raised when the [Visible](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract event EventHandler VisibleChanged
```

See Also

Reference

[IDrawable Interface](#)

[IDrawable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGameComponent Interface

Defines an interface for game components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public interface IGameComponent
```

See Also

Reference

[IGameComponent Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGameComponent Members

The following tables list the members exposed by the IGameComponent type.

Public Methods

	Name	Description
	Initialize	Called when the component should be initialized. This method can be used for tasks like querying for services the component needs and setting up non-graphics resources.

See Also

Reference

[IGameComponent Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IGameComponent Methods

Public Methods

	Name	Description
	Initialize	Called when the component should be initialized. This method can be used for tasks like querying for services the component needs and setting up non-graphics resources.

See Also

Reference

[IGameComponent Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IGameComponent.Initialize Method

Called when the component should be initialized. This method can be used for tasks like querying for services the component needs and setting up non-graphics resources.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Initialize ()
```

See Also

Reference

[IGameComponent Interface](#)

[IGameComponent Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceManager Interface

Defines the interface for an object that manages a [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public interface IGraphicsDeviceManager
```

See Also

Reference

[IGraphicsDeviceManager Members](#)




[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceManager Members

The following tables list the members exposed by the IGraphicsDeviceManager type.

Public Methods

	Name	Description
	BeginDraw	Starts the drawing of a frame.
	CreateDevice	Called to ensure that the device manager has created a valid device.
	EndDraw	Called by the game at the end of drawing; presents the final rendering.

See Also




Reference

[IGraphicsDeviceManager Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IGraphicsDeviceManager Methods

Public Methods

	Name	Description
	BeginDraw	Starts the drawing of a frame.
	CreateDevice	Called to ensure that the device manager has created a valid device.
	EndDraw	Called by the game at the end of drawing; presents the final rendering.

See Also

Reference

[IGraphicsDeviceManager Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IGraphicsDeviceManager.BeginDraw Method

Starts the drawing of a frame.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public bool BeginDraw ()
```

Return Value

true if the frame should be drawn; **false** otherwise.

See Also

Reference

[IGraphicsDeviceManager Interface](#)

[IGraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceManager.CreateDevice Method

Called to ensure that the device manager has created a valid device.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void CreateDevice ()
```

See Also

Reference

[IGraphicsDeviceManager Interface](#)

[IGraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceManager.EndDraw Method

Called by the game at the end of drawing; presents the final rendering.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void EndDraw ()
```

See Also

Reference

[IGraphicsDeviceManager Interface](#)

[IGraphicsDeviceManager Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IUpdateable Interface

Defines an interface for a game component that should be updated in [Game.Update](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public interface IUpdateable
```

See Also

Reference

[IUpdateable Members](#)



[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


IUpdateable Members

The following tables list the members exposed by the IUpdateable type.



Public Properties

	Name	Description
	Enabled	Indicates whether the game component's Update method should be called in Game.Update .
	UpdateOrder	Indicates when the game component should be updated relative to other game components. Lower values are updated first.

Public Methods

	Name	Description
	Update	Called when the game component should be updated.

Public Events

	Name	Description
	EnabledChanged	Raised when the Enabled property changes.
	UpdateOrderChanged	Raised when the UpdateOrder property changes.

See Also


Reference

[IUpdateable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IUpdateable Methods

Public Methods

	Name	Description
	Update	Called when the game component should be updated.

See Also

Reference

[IUpdateable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IUpdateable.Update Method

Called when the game component should be updated.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public void Update (  
    GameTime gameTime  
)
```

Parameters

gameTime

Snapshot of the game's timing state.

See Also

Reference

[IUpdateable Interface](#)



[IUpdateable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IUpdateable Properties

Public Properties

	Name	Description
	Enabled	Indicates whether the game component's Update method should be called in Game.Update .
	UpdateOrder	Indicates when the game component should be updated relative to other game components. Lower values are updated first.

See Also

Reference

[IUpdateable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IUpdateable.Enabled Property

Indicates whether the game component's [Update](#) method should be called in [Game.Update](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract bool Enabled { get; }
```

Property Value

true if [Update](#) should be called; **false** otherwise.

See Also

Reference

[IUpdateable Interface](#)

[IUpdateable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IUpdateable.UpdateOrder Property

Indicates when the game component should be updated relative to other game components. Lower values are updated first.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract int UpdateOrder { get; }
```

Property Value

When the game component should be updated.

See Also

Reference

[IUpdateable Interface](#)



[IUpdateable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IUpdateable Events

Public Events

	Name	Description
	EnabledChanged	Raised when the Enabled property changes.
	UpdateOrderChanged	Raised when the UpdateOrder property changes.

See Also

Reference

[IUpdateable Interface](#)

[Microsoft.Xna.Framework Namespace](#)

IUpdateable.EnabledChanged Event

Raised when the [Enabled](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract event EventHandler EnabledChanged
```

See Also

Reference

[IUpdateable Interface](#)

[IUpdateable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IUpdateable.UpdateOrderChanged Event

Raised when the [UpdateOrder](#) property changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public abstract event EventHandler UpdateOrderChanged
```

See Also

Reference

[IUpdateable Interface](#)

[IUpdateable Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper Class

Contains commonly used precalculated values.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class MathHelper
```

See Also

Reference

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide








[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






























MathHelper Members

The following tables list the members exposed by the MathHelper type.



Public Fields

	Name	Description
	E	Represents the mathematical constant e.
	Log10E	Represents the log base ten of e.
	Log2E	Represents the log base two of e.
	Pi	Represents the value of pi.
	PiOver2	Represents the value of pi divided by two.
	PiOver4	Represents the value of pi divided by four.
	TwoPi	Represents the value of pi times two.

Public Methods

	Name	Description
 	Barycentric	Returns the Cartesian coordinate for one axis of a point that is defined by a given triangle and two normalized barycentric (areal) coordinates.
 	CatmullRom	Performs a Catmull-Rom interpolation using the specified positions.
 	Clamp	Restricts a value to be within a specified range.
 	Distance	Calculates the absolute value of the difference of two values.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
 	Hermite	Performs a Hermite spline interpolation.
 	Lerp	Linearly interpolates between two values.
 	Max	Returns the greater of two values.
 	Min	Returns the lesser of two values.
	ReferenceEquals	(Inherited from Object .)
 	SmoothStep	Interpolates between two values using a cubic equation.
 	ToDegrees	Converts radians to degrees.
 	ToRadians	Converts degrees to radians.
	ToString	(Inherited from Object .)
 	WrapAngle	Reduces a given angle to a value between π and $-\pi$.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[MathHelper Class](#)

[Microsoft.Xna.Framework Namespace](#)

MathHelper Fields

Public Fields

	Name	Description
	E	Represents the mathematical constant e.
	Log10E	Represents the log base ten of e.
	Log2E	Represents the log base two of e.
	Pi	Represents the value of pi.
	PiOver2	Represents the value of pi divided by two.
	PiOver4	Represents the value of pi divided by four.
	TwoPi	Represents the value of pi times two.

See Also

Reference

[MathHelper Class](#)

[Microsoft.Xna.Framework Namespace](#)

MathHelper.E Field

Represents the mathematical constant e.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float E
```

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Log10E Field

Represents the log base ten of e.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float Log10E
```

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Log2E Field

Represents the log base two of e.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float Log2E
```

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Pi Field

Represents the value of pi.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float Pi
```

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.PiOver2 Field

Represents the value of pi divided by two.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float PiOver2
```

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.PiOver4 Field

Represents the value of pi divided by four.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float PiOver4
```

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.TwoPi Field

Represents the value of pi times two.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const float TwoPi
```

See Also

Reference

[MathHelper Class](#)


















[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper Methods

Public Methods

	Name	Description
	Barycentric	Returns the Cartesian coordinate for one axis of a point that is defined by a given triangle and two normalized barycentric (areal) coordinates.
	CatmullRom	Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Restricts a value to be within a specified range.
	Distance	Calculates the absolute value of the difference of two values.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Hermite	Performs a Hermite spline interpolation.
	Lerp	Linearly interpolates between two values.
	Max	Returns the greater of two values.
	Min	Returns the lesser of two values.
	ReferenceEquals	(Inherited from Object .)
	SmoothStep	Interpolates between two values using a cubic equation.
	ToDegrees	Converts radians to degrees.
	ToRadians	Converts degrees to radians.
	ToString	(Inherited from Object .)
	WrapAngle	Reduces a given angle to a value between π and $-\pi$.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MathHelper Class](#)

[Microsoft.Xna.Framework Namespace](#)

MathHelper.Barycentric Method

Returns the Cartesian coordinate for one axis of a point that is defined by a given triangle and two normalized barycentric (areal) coordinates.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Barycentric (
    float value1,
    float value2,
    float value3,
    float amount1,
    float amount2
)
```

Parameters

value1

The coordinate on one axis of vertex 1 of the defining triangle.

value2

The coordinate on the same axis of vertex 2 of the defining triangle.

value3

The coordinate on the same axis of vertex 3 of the defining triangle.

amount1

The normalized barycentric (areal) coordinate **b2**, equal to the weighting factor for vertex 2, the coordinate of which is specified in *value2*.

amount2

The normalized barycentric (areal) coordinate **b3**, equal to the weighting factor for vertex 3, the coordinate of which is specified in *value3*.

Return Value

Cartesian coordinate of the specified point with respect to the axis being used.

Remarks

About Barycentric Coordinates

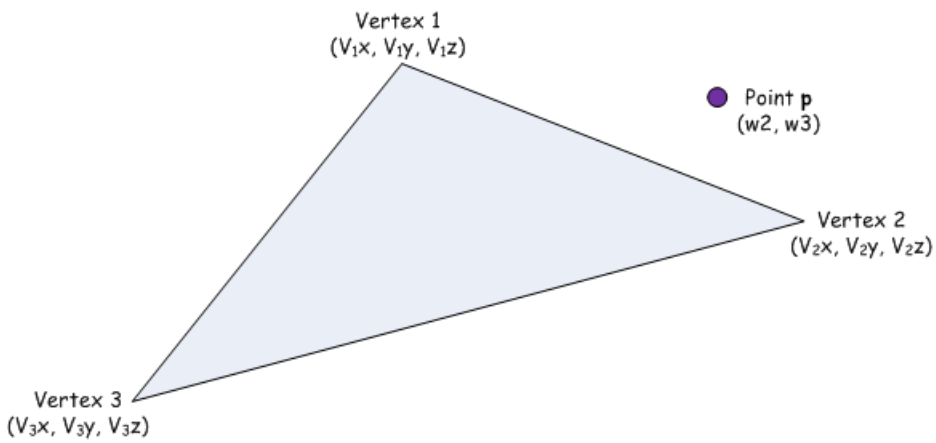
Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by three weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas.

$$\begin{aligned} P_x &= (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\ P_z &= (b_1 * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \end{aligned}$$

Such triple-weighting factors **b1**, **b2**, and **b3** are called *barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that **(k * b1)**, **(k * b2)**, and **(k * b3)** are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of barycentric coordinates is normalized so that **b1 + b2 + b3 = 1**, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** is known to equal **(1 - b2 - b3)**.



What MathHelper Barycentric Does

On an axis **a** (**a** being the x-, y-, or z-axis), the [MathHelper.Barycentric](#) method takes three triangle vertex coordinates on that axis (**V1a**, **V2a**, and **V3a**), and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument, and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

[Barycentric](#) then calculates the Cartesian coordinate of **Pa** as follows:

$$Pa = ((1 - b2 - b3) * V1a) + (b2 * V2a) + (b3 * V3a);$$

Thus, to calculate the x-axis Cartesian coordinate of **P**, you would pass the x-coordinates of the triangle vertices to [Barycentric](#) together with the appropriate areal coordinates of **P**.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.CatmullRom Method

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float CatmullRom (  
    float value1,  
    float value2,  
    float value3,  
    float value4,  
    float amount  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

Return Value

A position that is the result of the Catmull-Rom interpolation.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Clamp Method

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Clamp (  
    float value,  
    float min,  
    float max  
)
```

Parameters

value

The value to clamp.

min

The minimum value. If *value* is less than *min*, *min* will be returned.

max

The maximum value. If *value* is greater than *max*, *max* will be returned.

Return Value

The clamped value.

- If $value > max$, *max* will be returned.
- If $value < min$, *min* will be returned.
- If $min \leq value \leq max$, *value* will be returned.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Change Sound Volume Levels Using XACT](#)

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Distance Method

Calculates the absolute value of the difference of two values.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Distance (  
    float value1,  
    float value2  
)
```

Parameters

value1

Source value.

value2

Source value.

Return Value

Distance between the two values.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Hermite Method

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Hermite (  
    float value1,  
    float tangent1,  
    float value2,  
    float tangent2,  
    float amount  
)
```

Parameters

value1

Source position.

tangent1

Source tangent.

value2

Source position.

tangent2

Source tangent.

amount

Weighting factor.

Return Value

The result of the Hermite spline interpolation.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Lerp Method

Linearly interpolates between two values.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Lerp (  
    float value1,  
    float value2,  
    float amount  
)
```

Parameters

value1

Source value.

value2

Source value.

amount

Value between 0 and 1 indicating the weight of *value2*.

Return Value

Interpolated value.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned, a value of 1 will cause *value2* to be returned.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Max Method

Returns the greater of two values.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Max (  
    float value1,  
    float value2  
)
```

Parameters

value1

Source value.

value2

Source value.

Return Value

The greater value.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.Min Method

Returns the lesser of two values.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Min (  
    float value1,  
    float value2  
)
```

Parameters

value1

Source value.

value2

Source value.

Return Value

The lesser value.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.SmoothStep Method

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float SmoothStep (  
    float value1,  
    float value2,  
    float amount  
)
```

Parameters

value1

Source value.

value2

Source value.

amount

Weighting value.

Return Value

Interpolated value.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.ToDegrees Method

Converts radians to degrees.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float ToDegrees (  
    float radians  
)
```

Parameters

radians

The angle in radians.

Return Value

The angle in degrees.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.ToRadians Method

Converts degrees to radians.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float ToRadians (  
    float degrees  
)
```

Parameters

degrees

The angle in degrees.

Return Value

The angle in radians.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MathHelper.WrapAngle Method

Reduces a given angle to a value between π and $-\pi$.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float WrapAngle (  
    float angle  
)
```

Parameters

angle

The angle to reduce, in radians.

Return Value

The new angle, in radians.

See Also

Reference

[MathHelper Class](#)

[MathHelper Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix Structure

Defines a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.MatrixConverter)")]
[SerializableAttribute]
public struct Matrix : IEquatable<Matrix>
```

Remarks Matrices use a row vector layout in the XNA Framework. Matrices can be either row vector or column vector. Row vector matrices view vectors as a row from left to right, while column vector matrices view vectors as a column from top to bottom. For example, the x, y, and z of a matrix's translation vector in the XNA Framework would correspond to the fields [M41](#), [M42](#), [M43](#).

See Also

Reference

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

[How To: Position the Camera to View All Objects in a Scene](#)


[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

















Matrix Members

The following tables list the members exposed by the Matrix type.









Public Constructors

	Name	Description
	Matrix	Initializes a new instance of Matrix .










Public Fields





























































	Name	Description
	M11	Value at row 1 column 1 of the matrix.
	M12	Value at row 1 column 2 of the matrix.
	M13	Value at row 1 column 3 of the matrix.
	M14	Value at row 1 column 4 of the matrix.
	M21	Value at row 2 column 1 of the matrix.
	M22	Value at row 2 column 2 of the matrix.
	M23	Value at row 2 column 3 of the matrix.
	M24	Value at row 2 column 4 of the matrix.
	M31	Value at row 3 column 1 of the matrix.
	M32	Value at row 3 column 2 of the matrix.
	M33	Value at row 3 column 3 of the matrix.
	M34	Value at row 3 column 4 of the matrix.
	M41	Value at row 4 column 1 of the matrix.
	M42	Value at row 4 column 2 of the matrix.
	M43	Value at row 4 column 3 of the matrix.
	M44	Value at row 4 column 4 of the matrix.

Public Properties



	Name	Description
	Backward	Gets and sets the backward vector of the Matrix .
	Down	Gets and sets the down vector of the Matrix .
	Forward	Gets and sets the forward vector of the Matrix .
	Identity	Returns an instance of the identity matrix.
	Left	Gets and sets the left vector of the Matrix .
	Right	Gets and sets the right vector of the Matrix .
	Translation	Gets and sets the translation vector of the Matrix .
	Up	Gets and sets the up vector of the Matrix .

Public Methods

	Name	Description
	Add	Overloaded. Adds a matrix to another matrix.
	CreateBillboard	Overloaded. Creates a spherical billboard that rotates around a specified object position.
	CreateConstrainedBillboard	Overloaded. Creates a cylindrical billboard that rotates around a specified axis.
	CreateFromAxisAngle	Overloaded. Creates a new Matrix that rotates around an arbitrary vector.
	CreateFromQuaternion	Overloaded. Creates a rotation Matrix from a Quaternion .
	CreateFromYawPitchRoll	Overloaded. Creates a new rotation matrix from a specified yaw, pitch, and roll.
	CreateLookAt	Overloaded. Creates a view matrix.
	CreateOrthographic	Overloaded. Builds an orthogonal projection matrix.
	CreateOrthographicOffCenter	Overloaded. Builds a customized, orthogonal projection matrix.

 	CreatePerspective	Overloaded. Builds a perspective projection matrix.
 	CreatePerspectiveFieldOfView	Overloaded. Builds a perspective projection matrix based on a field of view.
 	CreatePerspectiveOffCenter	Overloaded. Builds a customized, perspective projection matrix.
 	CreateReflection	Overloaded. Creates a Matrix that reflects the coordinate system about a specified Plane .
 	CreateRotationX	Overloaded. Creates a matrix that can be used to rotate a set of vertices around the x-axis.
 	CreateRotationY	Overloaded. Creates a matrix that can be used to rotate a set of vertices around the y-axis.
 	CreateRotationZ	Overloaded. Creates a matrix that can be used to rotate a set of vertices around the z-axis.
 	CreateScale	Overloaded. Creates a scaling Matrix .
 	CreateShadow	Overloaded. Creates a Matrix that flattens geometry into a specified Plane as if casting a shadow from a specified light source.
 	CreateTranslation	Overloaded. Creates a translation Matrix .
 	CreateWorld	Overloaded. Creates a world matrix.
	Decompose	Extracts the scalar, translation, and rotation components from a 3D scale/rotate/translate (SRT) Matrix .
	Determinant	Calculates the determinant of the matrix.
 	Divide	Overloaded. Divides a matrix by a scalar value or the components of another matrix.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of this object.
	GetType	(Inherited from Object .)
 	Invert	Overloaded. Calculates the inverse of a matrix.
 	Lerp	Overloaded. Linearly interpolates between the corresponding values of two matrices.
 	Multiply	Overloaded. Multiplies a matrix by a scalar value or another matrix.
 	Negate	Overloaded. Negates individual elements of a matrix.
 	op_Addition	Adds a matrix to another matrix.
 	op_Division	Overloaded. Divides a matrix by a scalar value or the components of another matrix.
 	op_Equality	Compares a matrix for equality with another matrix.
 	op_Inequality	Tests a matrix for inequality with another matrix.
 	op_Multiply	Overloaded. Multiplies a matrix by a scalar value or another matrix.
 	op_Subtraction	Subtracts matrices.
 	op_UnaryNegation	Negates individual elements of a matrix.
	ReferenceEquals	(Inherited from Object .)
 	Subtract	Overloaded. Subtracts matrices.
 	ToString	Retrieves a string representation of the current object.
 	Transform	Overloaded. Transforms a Matrix by applying a Quaternion rotation.
 	Transpose	Overloaded. Transposes the rows and columns of a matrix.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

















Reference

[Matrix Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix Fields

Public Fields

	Name	Description
	M11	Value at row 1 column 1 of the matrix.
	M12	Value at row 1 column 2 of the matrix.
	M13	Value at row 1 column 3 of the matrix.
	M14	Value at row 1 column 4 of the matrix.
	M21	Value at row 2 column 1 of the matrix.
	M22	Value at row 2 column 2 of the matrix.
	M23	Value at row 2 column 3 of the matrix.
	M24	Value at row 2 column 4 of the matrix.
	M31	Value at row 3 column 1 of the matrix.
	M32	Value at row 3 column 2 of the matrix.
	M33	Value at row 3 column 3 of the matrix.
	M34	Value at row 3 column 4 of the matrix.
	M41	Value at row 4 column 1 of the matrix.
	M42	Value at row 4 column 2 of the matrix.
	M43	Value at row 4 column 3 of the matrix.
	M44	Value at row 4 column 4 of the matrix.

See Also

Reference

[Matrix Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.M11 Field

Value at row 1 column 1 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M11
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M12 Field

Value at row 1 column 2 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M12
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M13 Field

Value at row 1 column 3 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M13
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M14 Field

Value at row 1 column 4 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M14
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M21 Field

Value at row 2 column 1 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M21
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M22 Field

Value at row 2 column 2 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M22
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M23 Field

Value at row 2 column 3 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M23
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M24 Field

Value at row 2 column 4 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M24
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M31 Field

Value at row 3 column 1 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M31
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M32 Field

Value at row 3 column 2 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M32
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M33 Field

Value at row 3 column 3 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M33
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M34 Field

Value at row 3 column 4 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M34
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M41 Field

Value at row 4 column 1 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M41
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M42 Field

Value at row 4 column 2 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M42
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M43 Field

Value at row 4 column 3 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M43
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.M44 Field

Value at row 4 column 4 of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float M44
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix Constructor

Initializes a new instance of [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix (  
    float m11,  
    float m12,  
    float m13,  
    float m14,  
    float m21,  
    float m22,  
    float m23,  
    float m24,  
    float m31,  
    float m32,  
    float m33,  
    float m34,  
    float m41,  
    float m42,  
    float m43,  
    float m44  
)
```

Parameters

m11

Value to initialize *m11* to.

m12

Value to initialize *m12* to.

m13

Value to initialize *m13* to.

m14

Value to initialize *m14* to.

m21

Value to initialize *m21* to.

m22

Value to initialize *m22* to.

m23

Value to initialize *m23* to.

m24

Value to initialize *m24* to.

m31

Value to initialize *m31* to.

m32

Value to initialize *m32* to.

m33

Value to initialize *m33* to.

m34

Value to initialize *m34* to.

m41

Value to initialize *m41* to.

m42

Value to initialize *m42* to.

m43

Value to initialize *m43* to.

m44

Value to initialize *m44* to.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)














[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Matrix Methods

Public Methods

	Name	Description
	Add	Overloaded. Adds a matrix to another matrix.
	CreateBillboard	Overloaded. Creates a spherical billboard that rotates around a specified object position.
	CreateConstrainedBillboard	Overloaded. Creates a cylindrical billboard that rotates around a specified axis.
	CreateFromAxisAngle	Overloaded. Creates a new Matrix that rotates around an arbitrary vector.
	CreateFromQuaternion	Overloaded. Creates a rotation Matrix from a Quaternion .
	CreateFromYawPitchRoll	Overloaded. Creates a new rotation matrix from a specified yaw, pitch, and roll.
	CreateLookAt	Overloaded. Creates a view matrix.
	CreateOrthographic	Overloaded. Builds an orthogonal projection matrix.
	CreateOrthographicOffCenter	Overloaded. Builds a customized, orthogonal projection matrix.
	CreatePerspective	Overloaded. Builds a perspective projection matrix.
	CreatePerspectiveFieldOfView	Overloaded. Builds a perspective projection matrix based on a field of view.
	CreatePerspectiveOffCenter	Overloaded. Builds a customized, perspective projection matrix.
	CreateReflection	Overloaded. Creates a Matrix that reflects the coordinate system about a specified Plane .
	CreateRotationX	Overloaded. Creates a matrix that can be used to rotate a set of vertices around the x-axis.
	CreateRotationY	Overloaded. Creates a matrix that can be used to rotate a set of vertices around the y-axis.
	CreateRotationZ	Overloaded. Creates a matrix that can be used to rotate a set of vertices around the z-axis.
	CreateScale	Overloaded. Creates a scaling Matrix .
	CreateShadow	Overloaded. Creates a Matrix that flattens geometry into a specified Plane as if casting a shadow from a specified light source.
	CreateTranslation	Overloaded. Creates a translation Matrix .
	CreateWorld	Overloaded. Creates a world matrix.
	Decompose	Extracts the scalar, translation, and rotation components from a 3D scale/rotate/translate (SRT) Matrix .
	Determinant	Calculates the determinant of the matrix.
	Divide	Overloaded. Divides a matrix by a scalar value or the components of another matrix.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of this object.
	GetType	(Inherited from Object .)
	Invert	Overloaded. Calculates the inverse of a matrix.
	Lerp	Overloaded. Linearly interpolates between the corresponding values of two matrices.
	Multiply	Overloaded. Multiplies a matrix by a scalar value or another matrix.

	Negate	Overloaded. Negates individual elements of a matrix.
	op_Addition	Adds a matrix to another matrix.
	op_Division	Overloaded. Divides a matrix by a scalar value or the components of another matrix.
	op_Equality	Compares a matrix for equality with another matrix.
	op_Inequality	Tests a matrix for inequality with another matrix.
	op_Multiply	Overloaded. Multiplies a matrix by a scalar value or another matrix.
	op_Subtraction	Subtracts matrices.
	op_UnaryNegation	Negates individual elements of a matrix.
	ReferenceEquals	(Inherited from Object .)
	Subtract	Overloaded. Subtracts matrices.
	ToString	Retrieves a string representation of the current object.
	Transform	Overloaded. Transforms a Matrix by applying a Quaternion rotation.
	Transpose	Overloaded. Transposes the rows and columns of a matrix.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Matrix Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Add Method

Adds a matrix to another matrix.

Overload List

Name	Description
Matrix.Add (Matrix, Matrix)	Adds a matrix to another matrix.
Matrix.Add (Matrix, Matrix, Matrix)	Adds a matrix to another matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Add Method (Matrix, Matrix)

Adds a matrix to another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Add (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Add Method (Matrix, Matrix, Matrix)

Adds a matrix to another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Add (  
    ref Matrix matrix1,  
    ref Matrix matrix2,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

result

[[OutAttribute](#)] Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateBillboard Method

Creates a spherical billboard that rotates around a specified object position.

Overload List

Name	Description
Matrix.CreateBillboard (Vector3, Vector3, Vector3, Nullable<Vector3>)	Creates a spherical billboard that rotates around a specified object position.
Matrix.CreateBillboard (Vector3, Vector3, Vector3, Nullable<Vector3>, Matrix)	Creates a spherical billboard that rotates around a specified object position.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateBillboard Method (Vector3, Vector3, Vector3, Nullable<Vector3>)

Creates a spherical billboard that rotates around a specified object position.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateBillboard (  
    Vector3 objectPosition,  
    Vector3 cameraPosition,  
    Vector3 cameraUpVector,  
    Nullable<Vector3> cameraForwardVector  
)
```

Parameters

objectPosition

Position of the object the billboard will rotate around.

cameraPosition

Position of the camera.

cameraUpVector

The up vector of the camera.

cameraForwardVector

Optional forward vector of the camera.

Return Value

The created billboard matrix.

Remarks

This method computes the facing direction of the billboard from the object position and camera position. When the object and camera positions are too close, the matrix will not be accurate. To avoid this problem, the method uses the optional camera forward vector if the positions are too close.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateBillboard Method (Vector3, Vector3, Vector3, Nullable<Vector3>, Matrix)

Creates a spherical billboard that rotates around a specified object position.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateBillboard (  
    ref Vector3 objectPosition,  
    ref Vector3 cameraPosition,  
    ref Vector3 cameraUpVector,  
    Nullable<Vector3> cameraForwardVector,  
    out Matrix result  
)
```

Parameters

objectPosition

Position of the object the billboard will rotate around.

cameraPosition

Position of the camera.

cameraUpVector

The up vector of the camera.

cameraForwardVector

Optional forward vector of the camera.

result

[[OutAttribute](#)] The created billboard matrix.

Remarks

This method computes the facing direction of the billboard from the object position and camera position. When the object and camera positions are too close, the matrix will not be accurate. To avoid this problem, the method uses the optional camera forward vector if the positions are too close.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateConstrainedBillboard Method

Creates a cylindrical billboard that rotates around a specified axis.

Overload List

Name	Description
Matrix.CreateConstrainedBillboard (Vector3, Vector3, Vector3, Nullable<Vector3>, Nullable<Vector3>)	Creates a cylindrical billboard that rotates around a specified axis.
Matrix.CreateConstrainedBillboard (Vector3, Vector3, Vector3, Nullable<Vector3>, Nullable<Vector3>, Matrix)	Creates a cylindrical billboard that rotates around a specified axis.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateConstrainedBillboard Method (Vector3, Vector3, Vector3, Nullable<Vector3>, Nullable<Vector3>)

Creates a cylindrical billboard that rotates around a specified axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateConstrainedBillboard (  
    Vector3 objectPosition,  
    Vector3 cameraPosition,  
    Vector3 rotateAxis,  
    Nullable<Vector3> cameraForwardVector,  
    Nullable<Vector3> objectForwardVector  
)
```

Parameters

objectPosition

Position of the object the billboard will rotate around.

cameraPosition

Position of the camera.

rotateAxis

Axis to rotate the billboard around.

cameraForwardVector

Optional forward vector of the camera.

objectForwardVector

Optional forward vector of the object.

Return Value

The created billboard matrix.

Remarks

This method computes the facing direction of the billboard from the object position and camera position. When the object and camera positions are too close, the matrix will not be accurate. To avoid this problem, the method uses the optional camera forward vector if the positions are too close.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateConstrainedBillboard Method (Vector3, Vector3, Vector3, Nullable<Vector3>, Nullable<Vector3>, Matrix)

Creates a cylindrical billboard that rotates around a specified axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateConstrainedBillboard (  
    ref Vector3 objectPosition,  
    ref Vector3 cameraPosition,  
    ref Vector3 rotateAxis,  
    Nullable<Vector3> cameraForwardVector,  
    Nullable<Vector3> objectForwardVector,  
    out Matrix result  
)
```

Parameters

objectPosition

Position of the object the billboard will rotate around.

cameraPosition

Position of the camera.

rotateAxis

Axis to rotate the billboard around.

cameraForwardVector

Optional forward vector of the camera.

objectForwardVector

Optional forward vector of the object.

result

[[OutAttribute](#)] The created billboard matrix.

Remarks

This method computes the facing direction of the billboard from the object position and camera position. When the object and camera positions are too close, the matrix will not be accurate. To avoid this problem, the method uses the optional camera forward vector if the positions are too close.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateFromAxisAngle Method

Creates a new [Matrix](#) that rotates around an arbitrary vector.

Overload List

Name	Description
Matrix.CreateFromAxisAngle (Vector3, Single)	Creates a new Matrix that rotates around an arbitrary vector.
Matrix.CreateFromAxisAngle (Vector3, Single, Matrix)	Creates a new Matrix that rotates around an arbitrary vector.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateFromAxisAngle Method (Vector3, Single)

Creates a new [Matrix](#) that rotates around an arbitrary vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateFromAxisAngle (  
    Vector3 axis,  
    float angle  
)
```

Parameters

axis

The axis to rotate around.

angle

The angle to rotate around the vector.

Return Value

The created [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateFromAxisAngle Method (Vector3, Single, Matrix)

Creates a new [Matrix](#) that rotates around an arbitrary vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromAxisAngle (  
    ref Vector3 axis,  
    float angle,  
    out Matrix result  
)
```

Parameters

axis

The axis to rotate around.

angle

The angle to rotate around the vector.

result

[[OutAttribute](#)] The created [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateFromQuaternion Method

Creates a rotation [Matrix](#) from a [Quaternion](#).

Overload List

Name	Description
Matrix.CreateFromQuaternion (Quaternion)	Creates a rotation Matrix from a Quaternion .
Matrix.CreateFromQuaternion (Quaternion, Matrix)	Creates a rotation Matrix from a Quaternion .

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateFromQuaternion Method (Quaternion)

Creates a rotation [Matrix](#) from a [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateFromQuaternion (  
    Quaternion quaternion  
)
```

Parameters

quaternion

[Quaternion](#) to create the [Matrix](#) from.

Return Value

The created [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateFromQuaternion Method (Quaternion, Matrix)

Creates a rotation [Matrix](#) from a [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromQuaternion (  
    ref Quaternion quaternion,  
    out Matrix result  
)
```

Parameters

quaternion

[Quaternion](#) to create the [Matrix](#) from.

result

[[OutAttribute](#)] The created [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateFromYawPitchRoll Method

Creates a new rotation matrix from a specified yaw, pitch, and roll.

Overload List

Name	Description
Matrix.CreateFromYawPitchRoll (Single, Single, Single)	Creates a new rotation matrix from a specified yaw, pitch, and roll.
Matrix.CreateFromYawPitchRoll (Single, Single, Single, Matrix)	Fills in a rotation matrix from a specified yaw, pitch, and roll.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateFromYawPitchRoll Method (Single, Single, Single)

Creates a new rotation matrix from a specified yaw, pitch, and roll.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateFromYawPitchRoll (  
    float yaw,  
    float pitch,  
    float roll  
)
```

Parameters

yaw

Angle of rotation, in radians, around the y-axis.

pitch

Angle of rotation, in radians, around the x-axis.

roll

Angle of rotation, in radians, around the z-axis.

Return Value

A new rotation matrix with the specified yaw, pitch, and roll.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateFromYawPitchRoll Method (Single, Single, Single, Matrix)

Fills in a rotation matrix from a specified yaw, pitch, and roll.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromYawPitchRoll (  
    float yaw,  
    float pitch,  
    float roll,  
    out Matrix result  
)
```

Parameters

yaw

Angle of rotation, in radians, around the y-axis.

pitch

Angle of rotation, in radians, around the x-axis.

roll

Angle of rotation, in radians, around the z-axis.

result

[[OutAttribute](#)] An existing matrix filled in to represent the specified yaw, pitch, and roll.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateLookAt Method

Creates a view matrix.

Overload List

Name	Description
Matrix.CreateLookAt (Vector3, Vector3, Vector3)	Creates a view matrix.
Matrix.CreateLookAt (Vector3, Vector3, Vector3, Matrix)	Creates a view matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateLookAt Method (Vector3, Vector3, Vector3)

Creates a view matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateLookAt (  
    Vector3 cameraPosition,  
    Vector3 cameraTarget,  
    Vector3 cameraUpVector  
)
```

Parameters

cameraPosition

The position of the camera.

cameraTarget

The target towards which the camera is pointing.

cameraUpVector

The direction that is "up" from the camera's point of view.

Return Value

The created view matrix.

Remarks

View space, sometimes called camera space, is similar to world space in that it is typically used for the entire scene. However, in view space, the origin is at the viewer or camera.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateLookAt Method (Vector3, Vector3, Vector3, Matrix)

Creates a view matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateLookAt (  
    ref Vector3 cameraPosition,  
    ref Vector3 cameraTarget,  
    ref Vector3 cameraUpVector,  
    out Matrix result  
)
```

Parameters

cameraPosition

The position of the camera.

cameraTarget

The target towards which the camera is pointing.

cameraUpVector

The direction that is "up" from the camera's point of view.

result

[[OutAttribute](#)] The created view matrix.

Remarks

View space, sometimes called camera space, is similar to world space in that it is typically used for the entire scene. However, in view space, the origin is at the viewer or camera.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateOrthographic Method

Builds an orthogonal projection matrix.

Overload List

Name	Description
Matrix.CreateOrthographic (Single, Single, Single, Single)	Builds an orthogonal projection matrix.
Matrix.CreateOrthographic (Single, Single, Single, Single, Matrix)	Builds an orthogonal projection matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateOrthographic Method (Single, Single, Single, Single)

Builds an orthogonal projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateOrthographic (  
    float width,  
    float height,  
    float zNearPlane,  
    float zFarPlane  
)
```

Parameters

width

Width of the view volume.

height

Height of the view volume.

zNearPlane

Minimum z-value of the view volume.

zFarPlane

Maximum z-value of the view volume.

Return Value

The projection matrix.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y coordinates ranging from -1 to 1, and z coordinates ranging from 0 to 1.

Unlike perspective projection, in orthographic projection there is no perspective foreshortening.

The viewable area of this orthographic projection is centered on 0,0,0. The x-axis of the area ranges from $-width/2$ to $width/2$. The y-axis of the area ranges from $-height/2$ to $height/2$. The z-axis of the area ranges from *zNearPlane* to *zFarPlane*. These values are relative to the position and x, y, and z-axes of the view.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateOrthographic Method (Single, Single, Single, Single, Matrix)

Builds an orthogonal projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateOrthographic (  
    float width,  
    float height,  
    float zNearPlane,  
    float zFarPlane,  
    out Matrix result  
)
```

Parameters

width

Width of the view volume.

height

Height of the view volume.

zNearPlane

Minimum z-value of the view volume.

zFarPlane

Maximum z-value of the view volume.

result

[[OutAttribute](#)] The projection matrix.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y coordinates ranging from -1 to 1, and z coordinates ranging from 0 to 1.

Unlike perspective projection, in orthographic projection there is no perspective foreshortening.

The viewable area of this orthographic projection is centered on 0,0,0. The x-axis of the area ranges from $-width/2$ to $width/2$. The y-axis of the area ranges from $-height/2$ to $height/2$. The z-axis of the area ranges from *zNearPlane* to *zFarPlane*. These values are relative to the position and x, y, and z-axes of the view.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateOrthographicOffCenter Method

Builds a customized, orthogonal projection matrix.

Overload List

Name	Description
Matrix.CreateOrthographicOffCenter (Single, Single, Single, Single, Single, Single, Single)	Builds a customized, orthogonal projection matrix.
Matrix.CreateOrthographicOffCenter (Single, Single, Single, Single, Single, Single, Single, Matrix)	Builds a customized, orthogonal projection matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateOrthographicOffCenter Method (Single, Single, Single, Single, Single, Single)

Builds a customized, orthogonal projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateOrthographicOffCenter (
    float left,
    float right,
    float bottom,
    float top,
    float zNearPlane,
    float zFarPlane
)
```

Parameters

left

Minimum x-value of the view volume.

right

Maximum x-value of the view volume.

bottom

Minimum y-value of the view volume.

top

Maximum y-value of the view volume.

zNearPlane

Minimum z-value of the view volume.

zFarPlane

Maximum z-value of the view volume.

Return Value

The projection matrix.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y coordinates ranging from -1 to 1, and z coordinates ranging from 0 to 1.

Unlike perspective projection, in orthographic projection there is no perspective foreshortening.

The viewable area of this orthographic projection extends from *left* to *right* on the x-axis, *bottom* to *top* on the y-axis, and *zNearPlane* to *zFarPlane* on the z-axis. These values are relative to the position and x, y, and z-axes of the view.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateOrthographicOffCenter Method (Single, Single, Single, Single, Single, Matrix)

Builds a customized, orthogonal projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateOrthographicOffCenter (
    float left,
    float right,
    float bottom,
    float top,
    float zNearPlane,
    float zFarPlane,
    out Matrix result
)
```

Parameters

left

Minimum x-value of the view volume.

right

Maximum x-value of the view volume.

bottom

Minimum y-value of the view volume.

top

Maximum y-value of the view volume.

zNearPlane

Minimum z-value of the view volume.

zFarPlane

Maximum z-value of the view volume.

result

[OutAttribute] The projection matrix.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y coordinates ranging from -1 to 1, and z coordinates ranging from 0 to 1.

Unlike perspective projection, in orthographic projection there is no perspective foreshortening.

The viewable area of this orthographic projection extends from *left* to *right* on the x-axis, *bottom* to *top* on the y-axis, and *zNearPlane* to *zFarPlane* on the z-axis. These values are relative to the position and x, y, and z-axes of the view.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreatePerspective Method

Builds a perspective projection matrix.

Overload List

Name	Description
Matrix.CreatePerspective (Single, Single, Single, Single)	Builds a perspective projection matrix.
Matrix.CreatePerspective (Single, Single, Single, Single, Matrix)	Builds a perspective projection matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreatePerspective Method (Single, Single, Single, Single)

Builds a perspective projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreatePerspective (
    float width,
    float height,
    float nearPlaneDistance,
    float farPlaneDistance
)
```

Parameters

width

Width of the view volume at the near view plane.

height

Height of the view volume at the near view plane.

nearPlaneDistance

Distance to the near view plane.

farPlaneDistance

Distance to the far view plane.

Return Value

The projection matrix.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>nearPlaneDistance</i> or <i>farPlaneDistance</i> is a negative value. Specify a positive value for <i>nearPlaneDistance</i> or <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> larger than <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> must be smaller than <i>farPlaneDistance</i>.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y-coordinates ranging from -1 to 1, and a z-coordinate ranging from 0 to 1.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreatePerspective Method (Single, Single, Single, Single, Matrix)

Builds a perspective projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreatePerspective (
    float width,
    float height,
    float nearPlaneDistance,
    float farPlaneDistance,
    out Matrix result
)
```

Parameters

width

Width of the view volume at the near view plane.

height

Height of the view volume at the near view plane.

nearPlaneDistance

Distance to the near view plane.

farPlaneDistance

Distance to the far view plane.

result

[[OutAttribute](#)] The projection matrix.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>nearPlaneDistance</i> or <i>farPlaneDistance</i> is a negative value. Specify a positive value for <i>nearPlaneDistance</i> or <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> larger than <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> must be smaller than <i>farPlaneDistance</i>.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y-coordinates ranging from -1 to 1, and a z-coordinate ranging from 0 to 1.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreatePerspectiveFieldOfView Method

Builds a perspective projection matrix based on a field of view.

Overload List

Name	Description
Matrix.CreatePerspectiveFieldOfView (Single, Single, Single, Single)	Builds a perspective projection matrix based on a field of view.
Matrix.CreatePerspectiveFieldOfView (Single, Single, Single, Single, Matrix)	Builds a perspective projection matrix based on a field of view.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreatePerspectiveFieldOfView Method (Single, Single, Single, Single)

Builds a perspective projection matrix based on a field of view.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreatePerspectiveFieldOfView (
    float fieldOfView,
    float aspectRatio,
    float nearPlaneDistance,
    float farPlaneDistance
)
```

Parameters

fieldOfView

Field of view in radians.

aspectRatio

Aspect ratio, defined as view space width divided by height.

nearPlaneDistance

Distance to the near view plane.

farPlaneDistance

Distance to the far view plane.

Return Value

The perspective projection matrix.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>fieldOfView</i> is not between 0 and pi (180 degrees). Note that <i>fieldOfView</i> should be specified in radians. <i>nearPlaneDistance</i> or <i>farPlaneDistance</i> is a negative value. Specify a positive value for <i>nearPlaneDistance</i> or <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> larger than <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> must be smaller than <i>farPlaneDistance</i>.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y-coordinates ranging from -1 to 1, and a z-coordinate ranging from 0 to 1.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Concepts

[Viewports and Frustums](#)

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreatePerspectiveFieldOfView Method (Single, Single, Single, Single, Matrix)

Builds a perspective projection matrix based on a field of view.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreatePerspectiveFieldOfView (
    float fieldOfView,
    float aspectRatio,
    float nearPlaneDistance,
    float farPlaneDistance,
    out Matrix result
)
```

Parameters

fieldOfView

Field of view in radians.

aspectRatio

Aspect ratio, defined as view space width divided by height.

nearPlaneDistance

Distance to the near view plane.

farPlaneDistance

Distance to the far view plane.

result

[[OutAttribute](#)] The perspective projection matrix.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>fieldOfView</i> is not between 0 and pi (180 degrees). Note that <i>fieldOfView</i> should be specified in radians. <i>nearPlaneDistance</i> or <i>farPlaneDistance</i> is a negative value. Specify a positive value for <i>nearPlaneDistance</i> or <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> larger than <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> must be smaller than <i>farPlaneDistance</i>.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y-coordinates ranging from -1 to 1, and a z-coordinate ranging from 0 to 1.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Concepts

[Viewports and Frustums](#)

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreatePerspectiveOffCenter Method

Builds a customized, perspective projection matrix.

Overload List

Name	Description
Matrix.CreatePerspectiveOffCenter (Single, Single, Single, Single, Single, Single)	Builds a customized, perspective projection matrix.
Matrix.CreatePerspectiveOffCenter (Single, Single, Single, Single, Single, Single, Matrix)	Builds a customized, perspective projection matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreatePerspectiveOffCenter Method (Single, Single, Single, Single, Single, Single)

Builds a customized, perspective projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreatePerspectiveOffCenter (
    float left,
    float right,
    float bottom,
    float top,
    float nearPlaneDistance,
    float farPlaneDistance
)
```

Parameters

left

Minimum x-value of the view volume at the near view plane.

right

Maximum x-value of the view volume at the near view plane.

bottom

Minimum y-value of the view volume at the near view plane.

top

Maximum y-value of the view volume at the near view plane.

nearPlaneDistance

Distance to the near view plane.

farPlaneDistance

Distance to of the far view plane.

Return Value

The created projection matrix.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y-coordinates ranging from -1 to 1, and a z-coordinate ranging from 0 to 1.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>nearPlaneDistance</i> or <i>farPlaneDistance</i> is a negative value. Specify a positive value for <i>nearPlaneDistance</i> or <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> larger than <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> must be smaller than <i>farPlaneDistance</i>.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreatePerspectiveOffCenter Method (Single, Single, Single, Single, Single, Matrix)

Builds a customized, perspective projection matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreatePerspectiveOffCenter (
    float left,
    float right,
    float bottom,
    float top,
    float nearPlaneDistance,
    float farPlaneDistance,
    out Matrix result
)
```

Parameters

left

Minimum x-value of the view volume at the near view plane.

right

Maximum x-value of the view volume at the near view plane.

bottom

Minimum y-value of the view volume at the near view plane.

top

Maximum y-value of the view volume at the near view plane.

nearPlaneDistance

Distance to the near view plane.

farPlaneDistance

Distance to of the far view plane.

result

[[OutAttribute](#)] The created projection matrix.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>nearPlaneDistance</i> or <i>farPlaneDistance</i> is a negative value. Specify a positive value for <i>nearPlaneDistance</i> or <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> larger than <i>farPlaneDistance</i>. <i>nearPlaneDistance</i> must be smaller than <i>farPlaneDistance</i>.

Remarks

Projection space refers to the space after applying projection transformation from view space. After the projection transformation, visible content has x and y-coordinates ranging from -1 to 1, and a z-coordinate ranging from 0 to 1.

To obtain the viewable area (in world space) of a scene, create a [BoundingFrustum](#) and pass the combined view and projection matrix to the constructor.

See Also

Tasks

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateReflection Method

Creates a [Matrix](#) that reflects the coordinate system about a specified [Plane](#).

Overload List

Name	Description
Matrix.CreateReflection (Plane)	Creates a Matrix that reflects the coordinate system about a specified Plane .
Matrix.CreateReflection (Plane, Matrix)	Fills in an existing Matrix so that it reflects the coordinate system about a specified Plane .

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateReflection Method (Plane)

Creates a [Matrix](#) that reflects the coordinate system about a specified [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateReflection (  
    Plane value  
)
```

Parameters

value

The [Plane](#) about which to create a reflection.

Return Value

A new [Matrix](#) expressing the reflection.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateReflection Method (Plane, Matrix)

Fills in an existing [Matrix](#) so that it reflects the coordinate system about a specified [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateReflection (  
    ref Plane value,  
    out Matrix result  
)
```

Parameters

value

The [Plane](#) about which to create a reflection.

result

[[OutAttribute](#)] A [Matrix](#) that creates the reflection.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateRotationX Method

Creates a matrix that can be used to rotate a set of vertices around the x-axis.

Overload List

Name	Description
Matrix.CreateRotationX (Single)	Returns a matrix that can be used to rotate a set of vertices around the x-axis.
Matrix.CreateRotationX (Single, Matrix)	Populates data into a user-specified matrix that can be used to rotate a set of vertices around the x-axis.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateRotationX Method (Single)

Returns a matrix that can be used to rotate a set of vertices around the x-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateRotationX (  
    float radians  
)
```

Parameters

radians

The amount, in radians, in which to rotate around the x-axis. Note that you can use [ToRadians](#) to convert degrees to radians.

Return Value

The rotation matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateRotationX Method (Single, Matrix)

Populates data into a user-specified matrix that can be used to rotate a set of vertices around the x-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateRotationX (  
    float radians,  
    out Matrix result  
)
```

Parameters

radians

The amount, in radians, in which to rotate around the x-axis. Note that you can use [ToRadians](#) to convert degrees to radians.

result

[[OutAttribute](#)] The matrix in which to place the calculated data.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateRotationY Method

Creates a matrix that can be used to rotate a set of vertices around the y-axis.

Overload List

Name	Description
Matrix.CreateRotationY (Single)	Returns a matrix that can be used to rotate a set of vertices around the y-axis.
Matrix.CreateRotationY (Single, Matrix)	Populates data into a user-specified matrix that can be used to rotate a set of vertices around the y-axis.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateRotationY Method (Single)

Returns a matrix that can be used to rotate a set of vertices around the y-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateRotationY (  
    float radians  
)
```

Parameters

radians

The amount, in radians, in which to rotate around the y-axis. Note that you can use [ToRadians](#) to convert degrees to radians.

Return Value

The rotation matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateRotationY Method (Single, Matrix)

Populates data into a user-specified matrix that can be used to rotate a set of vertices around the y-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateRotationY (  
    float radians,  
    out Matrix result  
)
```

Parameters

radians

The amount, in radians, in which to rotate around the y-axis. Note that you can use [ToRadians](#) to convert degrees to radians.

result

[[OutAttribute](#)] The matrix in which to place the calculated data.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateRotationZ Method

Creates a matrix that can be used to rotate a set of vertices around the z-axis.

Overload List

Name	Description
Matrix.CreateRotationZ (Single)	Returns a matrix that can be used to rotate a set of vertices around the z-axis.
Matrix.CreateRotationZ (Single, Matrix)	Populates data into a user-specified matrix that can be used to rotate a set of vertices around the z-axis.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateRotationZ Method (Single)

Returns a matrix that can be used to rotate a set of vertices around the z-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateRotationZ (  
    float radians  
)
```

Parameters

radians

The amount, in radians, in which to rotate around the z-axis. Note that you can use [ToRadians](#) to convert degrees to radians.

Return Value

The rotation matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateRotationZ Method (Single, Matrix)

Populates data into a user-specified matrix that can be used to rotate a set of vertices around the z-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateRotationZ (  
    float radians,  
    out Matrix result  
)
```

Parameters

radians

The amount, in radians, in which to rotate around the z-axis. Note that you can use [ToRadians](#) to convert degrees to radians.

result

[[OutAttribute](#)] The rotation matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateScale Method

Creates a scaling [Matrix](#).

Overload List

Name	Description
Matrix.CreateScale (Single)	Creates a scaling Matrix .
Matrix.CreateScale (Single, Matrix)	Creates a scaling Matrix .
Matrix.CreateScale (Single, Single, Single)	Creates a scaling Matrix .
Matrix.CreateScale (Single, Single, Single, Matrix)	Creates a scaling Matrix .
Matrix.CreateScale (Vector3)	Creates a scaling Matrix .
Matrix.CreateScale (Vector3, Matrix)	Creates a scaling Matrix .

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

Matrix.CreateScale Method (Single)

Creates a scaling [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateScale (  
    float scale  
)
```

Parameters

scale

Amount to scale by.

Return Value

The created scaling [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateScale Method (Single, Matrix)

Creates a scaling [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateScale (  
    float scale,  
    out Matrix result  
)
```

Parameters

scale

Value to scale by.

result

[[OutAttribute](#)] The created scaling [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateScale Method (Single, Single, Single)

Creates a scaling [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateScale (  
    float xScale,  
    float yScale,  
    float zScale  
)
```

Parameters

xScale

Value to scale by on the x-axis.

yScale

Value to scale by on the y-axis.

zScale

Value to scale by on the z-axis.

Return Value

The created scaling [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateScale Method (Single, Single, Single, Matrix)

Creates a scaling [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateScale (  
    float xScale,  
    float yScale,  
    float zScale,  
    out Matrix result  
)
```

Parameters

xScale

Value to scale by on the x-axis.

yScale

Value to scale by on the y-axis.

zScale

Value to scale by on the z-axis.

result

[[OutAttribute](#)] The created scaling [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateScale Method (Vector3)

Creates a scaling [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateScale (  
    Vector3 scales  
)
```

Parameters

scales

Amounts to scale by on the x, y, and z axes.

Return Value

The created scaling [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateScale Method (Vector3, Matrix)

Creates a scaling [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateScale (  
    ref Vector3 scales,  
    out Matrix result  
)
```

Parameters

scales

Amounts to scale by on the x, y, and z axes.

result

[[OutAttribute](#)] The created scaling [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateShadow Method

Creates a [Matrix](#) that flattens geometry into a specified [Plane](#) as if casting a shadow from a specified light source.

Overload List

Name	Description
Matrix.CreateShadow (Vector3, Plane)	Creates a Matrix that flattens geometry into a specified Plane as if casting a shadow from a specified light source.
Matrix.CreateShadow (Vector3, Plane, Matrix)	Fills in a Matrix to flatten geometry into a specified Plane as if casting a shadow from a specified light source.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateShadow Method (Vector3, Plane)

Creates a [Matrix](#) that flattens geometry into a specified [Plane](#) as if casting a shadow from a specified light source.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateShadow (
    Vector3 lightDirection,
    Plane plane
)
```

Parameters

lightDirection

A [Vector3](#) specifying the direction from which the light that will cast the shadow is coming.

plane

The [Plane](#) onto which the new matrix should flatten geometry so as to cast a shadow.

Return Value

A new [Matrix](#) that can be used to flatten geometry onto the specified plane from the specified direction.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateShadow Method (Vector3, Plane, Matrix)

Fills in a [Matrix](#) to flatten geometry into a specified [Plane](#) as if casting a shadow from a specified light source.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateShadow (  
    ref Vector3 lightDirection,  
    ref Plane plane,  
    out Matrix result  
)
```

Parameters

lightDirection

A [Vector3](#) specifying the direction from which the light that will cast the shadow is coming.

plane

The [Plane](#) onto which the new matrix should flatten geometry so as to cast a shadow.

result

[[OutAttribute](#)] A [Matrix](#) that can be used to flatten geometry onto the specified plane from the specified direction.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateTranslation Method

Creates a translation [Matrix](#).

Overload List

Name	Description
Matrix.CreateTranslation (Single, Single, Single)	Creates a translation Matrix .
Matrix.CreateTranslation (Single, Single, Single, Matrix)	Creates a translation Matrix .
Matrix.CreateTranslation (Vector3)	Creates a translation Matrix .
Matrix.CreateTranslation (Vector3, Matrix)	Creates a translation Matrix .

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Matrix.CreateTranslation Method (Single, Single, Single)

Creates a translation [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateTranslation (
    float xPosition,
    float yPosition,
    float zPosition
)
```

Parameters

xPosition

Value to translate by on the x-axis.

yPosition

Value to translate by on the y-axis.

zPosition

Value to translate by on the z-axis.

Return Value

The created translation [Matrix](#).

See Also

Tasks

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateTranslation Method (Single, Single, Single, Matrix)

Creates a translation [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateTranslation (  
    float xPosition,  
    float yPosition,  
    float zPosition,  
    out Matrix result  
)
```

Parameters

xPosition

Value to translate by on the x-axis.

yPosition

Value to translate by on the y-axis.

zPosition

Value to translate by on the z-axis.

result

[[OutAttribute](#)] The created translation [Matrix](#).

See Also

Tasks

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateTranslation Method (Vector3)

Creates a translation [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateTranslation (  
    Vector3 position  
)
```

Parameters

position

Amounts to translate by on the x, y, and z axes.

Return Value

The created translation [Matrix](#).

See Also

Tasks

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateTranslation Method (Vector3, Matrix)

Creates a translation [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateTranslation (  
    ref Vector3 position,  
    out Matrix result  
)
```

Parameters

position

Amounts to translate by on the x, y, and z axes.

result

[[OutAttribute](#)] The created translation [Matrix](#).

See Also

Tasks

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateWorld Method

Creates a world matrix.

Overload List

Name	Description
Matrix.CreateWorld (Vector3, Vector3, Vector3)	Creates a world matrix with the specified parameters.
Matrix.CreateWorld (Vector3, Vector3, Vector3, Matrix)	Creates a world matrix with the specified parameters.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.CreateWorld Method (Vector3, Vector3, Vector3)

Creates a world matrix with the specified parameters.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix CreateWorld (  
    Vector3 position,  
    Vector3 forward,  
    Vector3 up  
)
```

Parameters

position

Position of the object. This value is used in translation operations.

forward

Forward direction of the object.

up

Upward direction of the object; usually [0, 1, 0].

Return Value

The created world matrix.

Remarks

This matrix includes rotation and translation, but not scaling.

See Also

Reference

[Matrix Structure](#)

[CreateScale](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.CreateWorld Method (Vector3, Vector3, Vector3, Matrix)

Creates a world matrix with the specified parameters.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateWorld (  
    ref Vector3 position,  
    ref Vector3 forward,  
    ref Vector3 up,  
    out Matrix result  
)
```

Parameters

position

Position of the object. This value is used in translation operations.

forward

Forward direction of the object.

up

Upward direction of the object; usually [0, 1, 0].

result

[[OutAttribute](#)] The created world matrix.

Remarks

This matrix includes rotation and translation, but not scaling.

See Also

Reference

[Matrix Structure](#)

[CreateScale](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Decompose Method

Extracts the scalar, translation, and rotation components from a 3D scale/rotate/translate (SRT) [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Decompose (  
    out Vector3 scale,  
    out Quaternion rotation,  
    out Vector3 translation  
)
```

Parameters

scale

[[OutAttribute](#)] The scalar component of the transform matrix, expressed as a [Vector3](#).

rotation

[[OutAttribute](#)] The rotation component of the transform matrix, expressed as a [Quaternion](#).

translation

[[OutAttribute](#)] The translation component of the transform matrix, expressed as a [Vector3](#).

Return Value

true if the [Matrix](#) can be decomposed; **false** otherwise.

Example The following example illustrates how to recombine the matrix from the extracted components.

```
Matrix m = Matrix.CreateScale(scale) *  
            Matrix.CreateFromQuaternion(rotation) *  
            Matrix.CreateTranslation(translation);
```

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Determinant Method

Calculates the determinant of the matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Determinant ()
```

Return Value

The determinant of the matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Divide Method

Divides a matrix by a scalar value or the components of another matrix.

Overload List

Name	Description
Matrix.Divide (Matrix, Matrix)	Divides the components of a matrix by the corresponding components of another matrix.
Matrix.Divide (Matrix, Matrix, Matrix)	Divides the components of a matrix by the corresponding components of another matrix.
Matrix.Divide (Matrix, Single)	Divides the components of a matrix by a scalar.
Matrix.Divide (Matrix, Single, Matrix)	Divides the components of a matrix by a scalar.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Divide Method (Matrix, Matrix)

Divides the components of a matrix by the corresponding components of another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Divide (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

The divisor.

Return Value

Result of the division.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Divide Method (Matrix, Matrix, Matrix)

Divides the components of a matrix by the corresponding components of another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Matrix matrix1,  
    ref Matrix matrix2,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

matrix2

The divisor.

result

[[OutAttribute](#)] Result of the division.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Divide Method (Matrix, Single)

Divides the components of a matrix by a scalar.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Divide (  
    Matrix matrix1,  
    float divider  
)
```

Parameters

matrix1

Source matrix.

divider

The divisor.

Return Value

Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Divide Method (Matrix, Single, Matrix)

Divides the components of a matrix by a scalar.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Matrix matrix1,  
    float divider,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

divider

The divisor.

result

[[OutAttribute](#)] Result of the division.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Matrix.Equals (Matrix)	Determines whether the specified Object is equal to the Matrix .
Matrix.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Matrix.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Equals Method (Matrix)

Determines whether the specified [Object](#) is equal to the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Matrix other  
)
```

Parameters

other

The [Object](#) to compare with the current [Matrix](#).

Return Value

true if the specified [Object](#) is equal to the current [Matrix](#); **false** otherwise.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.GetHashCode Method

Gets the hash code of this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code of this object.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Invert Method

Calculates the inverse of a matrix.

Overload List

Name	Description
Matrix.Invert (Matrix)	Calculates the inverse of a matrix.
Matrix.Invert (Matrix, Matrix)	Calculates the inverse of a matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Invert Method (Matrix)

Calculates the inverse of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Invert (  
    Matrix matrix  
)
```

Parameters

matrix

Source matrix.

Return Value

The inverse of the matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Invert Method (Matrix, Matrix)

Calculates the inverse of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Invert (  
    ref Matrix matrix,  
    out Matrix result  
)
```

Parameters

matrix

The source matrix.

result

[[OutAttribute](#)] The inverse of *matrix*. The same matrix can be used for both arguments.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Lerp Method

Linearly interpolates between the corresponding values of two matrices.

Overload List

Name	Description
Matrix.Lerp (Matrix, Matrix, Single)	Linearly interpolates between the corresponding values of two matrices.
Matrix.Lerp (Matrix, Matrix, Single, Matrix)	Linearly interpolates between the corresponding values of two matrices.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Lerp Method (Matrix, Matrix, Single)

Linearly interpolates between the corresponding values of two matrices.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Lerp (  
    Matrix matrix1,  
    Matrix matrix2,  
    float amount  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

amount

Interpolation value.

Return Value

Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Lerp Method (Matrix, Matrix, Single, Matrix)

Linearly interpolates between the corresponding values of two matrices.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Lerp (  
    ref Matrix matrix1,  
    ref Matrix matrix2,  
    float amount,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

amount

Interpolation value.

result

[[OutAttribute](#)] Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Multiply Method

Multiplies a matrix by a scalar value or another matrix.

Overload List

Name	Description
Matrix.Multiply (Matrix, Matrix)	Multiplies a matrix by another matrix.
Matrix.Multiply (Matrix, Matrix, Matrix)	Multiplies a matrix by another matrix.
Matrix.Multiply (Matrix, Single)	Multiplies a matrix by a scalar value.
Matrix.Multiply (Matrix, Single, Matrix)	Multiplies a matrix by a scalar value.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Multiply Method (Matrix, Matrix)

Multiplies a matrix by another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Multiply (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

Result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Multiply Method (Matrix, Matrix, Matrix)

Multiplies a matrix by another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Matrix matrix1,  
    ref Matrix matrix2,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

result

[[OutAttribute](#)] Result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Multiply Method (Matrix, Single)

Multiplies a matrix by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Multiply (  
    Matrix matrix1,  
    float scaleFactor  
)
```

Parameters

matrix1

Source matrix.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Multiply Method (Matrix, Single, Matrix)

Multiplies a matrix by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Matrix matrix1,  
    float scaleFactor,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

scaleFactor

Scalar value.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Negate Method

Negates individual elements of a matrix.

Overload List

Name	Description
Matrix.Negate (Matrix)	Negates individual elements of a matrix.
Matrix.Negate (Matrix, Matrix)	Negates individual elements of a matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Negate Method (Matrix)

Negates individual elements of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Negate (  
    Matrix matrix  
)
```

Parameters

matrix

Source matrix.

Return Value

Negated matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Negate Method (Matrix, Matrix)

Negates individual elements of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Negate (  
    ref Matrix matrix,  
    out Matrix result  
)
```

Parameters

matrix

Source matrix.

result

[[OutAttribute](#)] Negated matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Addition Method

Adds a matrix to another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Addition (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Division Method

Divides a matrix by a scalar value or the components of another matrix.

Overload List

Name	Description
Matrix.op_Division (Matrix, Matrix)	Divides the components of a matrix by the corresponding components of another matrix.
Matrix.op_Division (Matrix, Single)	Divides the components of a matrix by a scalar.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.op_Division Method (Matrix, Matrix)

Divides the components of a matrix by the corresponding components of another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Division (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

The divisor.

Return Value

Resulting matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Division Method (Matrix, Single)

Divides the components of a matrix by a scalar.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Division (  
    Matrix matrix1,  
    float divider  
)
```

Parameters

matrix1

Source matrix.

divider

The divisor.

Return Value

The result of the division.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Equality Method

Compares a matrix for equality with another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

true if the matrices are equal; **false** otherwise.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Inequality Method

Tests a matrix for inequality with another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

The matrix on the left of the equal sign.

matrix2

The matrix on the right of the equal sign.

Return Value

true if the matrices are not equal; **false** otherwise.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Multiply Method

Multiplies a matrix by a scalar value or another matrix.

Overload List

Name	Description
Matrix.op_Multiply (Matrix, Matrix)	Multiplies a matrix by another matrix.
Matrix.op_Multiply (Matrix, Single)	Multiplies a matrix by a scalar value.
Matrix.op_Multiply (Single, Matrix)	Multiplies a matrix by a scalar value.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.op_Multiply Method (Matrix, Matrix)

Multiplies a matrix by another matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Multiply (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

Result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Multiply Method (Matrix, Single)

Multiplies a matrix by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Multiply (  
    Matrix matrix,  
    float scaleFactor  
)
```

Parameters

matrix

Source matrix.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Multiply Method (Single, Matrix)

Multiplies a matrix by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Multiply (  
    float scaleFactor,  
    Matrix matrix  
)
```

Parameters

scaleFactor

Scalar value.

matrix

Source matrix.

Return Value

Result of the multiplication.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_Subtraction Method

Subtracts matrices.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_Subtraction (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

Result of the subtraction.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.op_UnaryNegation Method

Negates individual elements of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix op_UnaryNegation (  
    Matrix matrix1  
)
```

Parameters

matrix1

Source matrix.

Return Value

Negated matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Subtract Method

Subtracts matrices.

Overload List

Name	Description
Matrix.Subtract (Matrix, Matrix)	Subtracts matrices.
Matrix.Subtract (Matrix, Matrix, Matrix)	Subtracts matrices.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Subtract Method (Matrix, Matrix)

Subtracts matrices.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Subtract (  
    Matrix matrix1,  
    Matrix matrix2  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

Return Value

Result of the subtraction.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Subtract Method (Matrix, Matrix, Matrix)

Subtracts matrices.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Subtract (  
    ref Matrix matrix1,  
    ref Matrix matrix2,  
    out Matrix result  
)
```

Parameters

matrix1

Source matrix.

matrix2

Source matrix.

result

[[OutAttribute](#)] Result of the subtraction.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.ToString Method

Retrieves a string representation of the current object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Transform Method

Transforms a [Matrix](#) by applying a [Quaternion](#) rotation.

Overload List

Name	Description
Matrix.Transform (Matrix, Quaternion)	Transforms a Matrix by applying a Quaternion rotation.
Matrix.Transform (Matrix, Quaternion, Matrix)	Transforms a Matrix by applying a Quaternion rotation.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Transform Method (Matrix, Quaternion)

Transforms a [Matrix](#) by applying a [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Transform (  
    Matrix value,  
    Quaternion rotation  
)
```

Parameters

value

The [Matrix](#) to transform.

rotation

The rotation to apply, expressed as a [Quaternion](#).

Return Value

A new [Matrix](#) that is the result of the transform.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Transform Method (Matrix, Quaternion, Matrix)

Transforms a [Matrix](#) by applying a [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Matrix value,  
    ref Quaternion rotation,  
    out Matrix result  
)
```

Parameters

value

The [Matrix](#) to transform.

rotation

The rotation to apply, expressed as a [Quaternion](#).

result

[OutAttribute] An existing [Matrix](#) filled in with the result of the transform.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Transpose Method

Transposes the rows and columns of a matrix.

Overload List

Name	Description
Matrix.Transpose (Matrix)	Transposes the rows and columns of a matrix.
Matrix.Transpose (Matrix, Matrix)	Transposes the rows and columns of a matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Transpose Method (Matrix)

Transposes the rows and columns of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Transpose (  
    Matrix matrix  
)
```

Parameters

matrix

Source matrix.

Return Value

Transposed matrix.

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Transpose Method (Matrix, Matrix)

Transposes the rows and columns of a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transpose (  
    ref Matrix matrix,  
    out Matrix result  
)
```

Parameters

matrix

Source matrix.

result

[[OutAttribute](#)] Transposed matrix.

See Also

Reference

[Matrix Structure](#)









[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix Properties

Public Properties

	Name	Description
	Backward	Gets and sets the backward vector of the Matrix .
	Down	Gets and sets the down vector of the Matrix .
	Forward	Gets and sets the forward vector of the Matrix .
 S	Identity	Returns an instance of the identity matrix.
	Left	Gets and sets the left vector of the Matrix .
	Right	Gets and sets the right vector of the Matrix .
	Translation	Gets and sets the translation vector of the Matrix .
	Up	Gets and sets the up vector of the Matrix .

See Also

Reference

[Matrix Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Matrix.Backward Property

Gets and sets the backward vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Backward { get; set; }
```

Property Value

The backward vector of the [Matrix](#).

The default value is composed of the current values for the matrix elements in the third row: [M31](#), [M32](#), and [M33](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Position the Camera to View All Objects in a Scene](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Down Property

Gets and sets the down vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Down { get; set; }
```

Property Value

The down vector of the [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Forward Property

Gets and sets the forward vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Forward { get; set; }
```

Property Value

The forward vector of the [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Identity Property

Returns an instance of the identity matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Matrix Identity { get; }
```

Property Value

The identity matrix.

Remarks

Except for the main diagonal, whose elements are all ones, the identity matrix is a square matrix composed of zeros. Unlike all other matrix multiplications, the multiplication between an identity matrix and another matrix is commutative.

This is useful for initializing a matrix to a known value before you perform additional operations, such as a transformation.

Note

Multiplying a matrix with its inverse results in an identity matrix.

See Also

Reference

[Invert](#)

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Left Property

Gets and sets the left vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Left { get; set; }
```

Property Value

The left vector of the [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Right Property

Gets and sets the right vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Right { get; set; }
```

Property Value

The right vector of the [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Translation Property

Gets and sets the translation vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Translation { get; set; }
```

Property Value

The translation vector of the [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Matrix.Up Property

Gets and sets the up vector of the [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Up { get; set; }
```

Property Value

The up vector of the [Matrix](#).

See Also

Reference

[Matrix Structure](#)

[Matrix Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NoSuitableGraphicsDeviceException Class

Thrown when no available graphics device fits the given device preferences.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class NoSuitableGraphicsDeviceException : ApplicationException
```

See Also

Reference

[NoSuitableGraphicsDeviceException Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








NoSuitableGraphicsDeviceException Members

The following tables list the members exposed by the NoSuitableGraphicsDeviceException type.


Public Constructors

Name	Description
 NoSuitableGraphicsDeviceException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NoSuitableGraphicsDeviceException Class](#)

[Microsoft.Xna.Framework Namespace](#)

NoSuitableGraphicsDeviceException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
NoSuitableGraphicsDeviceException (String)	Initializes a new instance of this class with a specified error message.
NoSuitableGraphicsDeviceException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[NoSuitableGraphicsDeviceException Class](#)

[NoSuitableGraphicsDeviceException Members](#)

[Microsoft.Xna.Framework Namespace](#)

NoSuitableGraphicsDeviceException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public NoSuitableGraphicsDeviceException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[NoSuitableGraphicsDeviceException Class](#)

[NoSuitableGraphicsDeviceException Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NoSuitableGraphicsDeviceException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public NoSuitableGraphicsDeviceException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[NoSuitableGraphicsDeviceException Class](#)







[NoSuitableGraphicsDeviceException Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NoSuitableGraphicsDeviceException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[NoSuitableGraphicsDeviceException Class](#)


[Microsoft.Xna.Framework Namespace](#)

NoSuitableGraphicsDeviceException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[NoSuitableGraphicsDeviceException Class](#)

[Microsoft.Xna.Framework Namespace](#)

Plane Structure

Defines a plane.

Namespace: Microsoft.Xna.Framework

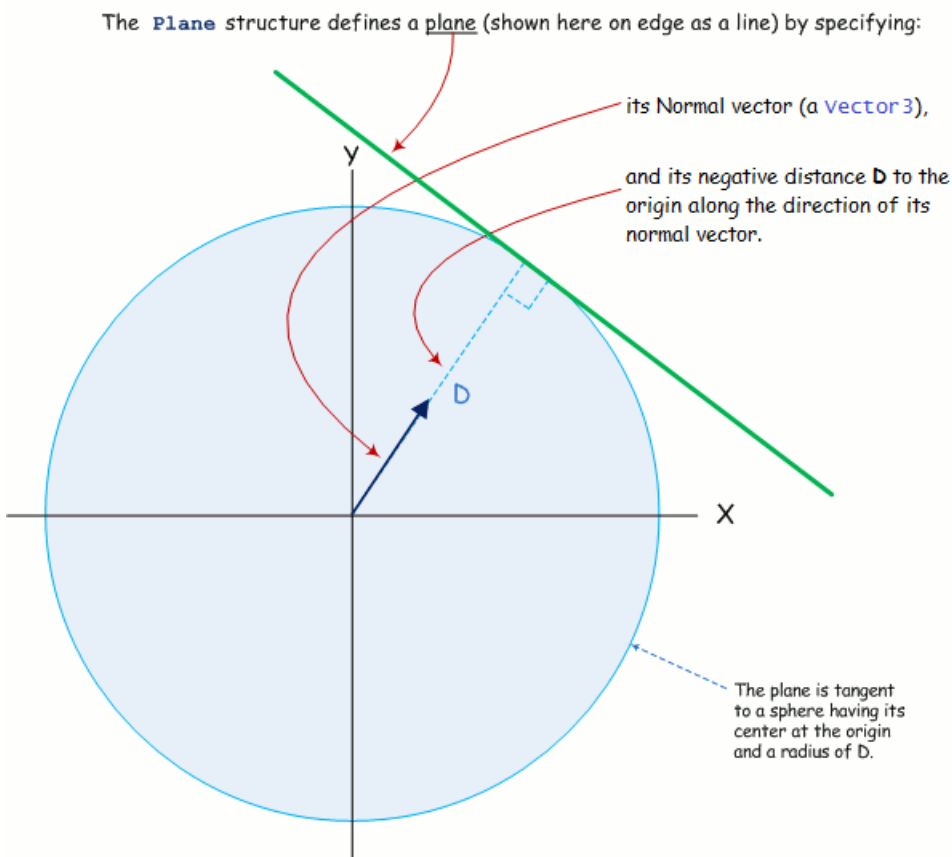
Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

```
C#
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.PlaneConverter)")]
[SerializableAttribute]
public struct Plane : IEquatable<Plane>
```

Remarks

The [Plane](#) structure represents a plane using a 3D vector normal and a distance value (stored as a negative value), as the following figure shows.



See Also

Reference

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Plane Members

The following tables list the members exposed by the Plane type.














Public Constructors

	Name	Description
	Plane	Overloaded. Creates an instance of Plane .



Public Fields

	Name	Description
	D	The distance of the Plane along its normal from the origin.
	Normal	The normal vector of the Plane .

Public Methods

	Name	Description
	Dot	Overloaded. Calculates the dot product of a specified Vector4 and this Plane .
	DotCoordinate	Overloaded. Returns the dot product of a specified Vector3 and the Normal vector of this Plane plus the D constant value of the Plane .
	DotNormal	Overloaded. Returns the dot product of a specified Vector3 and the Normal vector of this Plane .
	Equals	Overloaded. Determines whether two instances of Plane are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether a Plane intersects a bounding volume.
	Normalize	Overloaded. Changes the coefficients of the Normal vector of a Plane to make it of unit length.
	op_Equality	Determines whether two instances of Plane are equal.
	op_Inequality	Determines whether two instances of Plane are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current Plane .
	Transform	Overloaded. Transforms a normalized Plane by a Matrix or Quaternion .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[Plane Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Plane Fields

Public Fields

	Name	Description
	D	The distance of the Plane along its normal from the origin.
	Normal	The normal vector of the Plane .

See Also

Reference

[Plane Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.D Field

The distance of the [Plane](#) along its normal from the origin.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float D
```

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Normal Field

The normal [vector](#) of the [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Normal
```

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane Constructor

Creates an instance of [Plane](#).

Overload List

Name	Description
Plane (Single, Single, Single, Single)	Creates a new instance of Plane .
Plane (Vector3, Single)	Creates a new instance of Plane .
Plane (Vector3, Vector3, Vector3)	Creates a new instance of Plane .
Plane (Vector4)	Creates a new instance of Plane .

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane Constructor (Single, Single, Single, Single)

Creates a new instance of [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane (  
    float a,  
    float b,  
    float c,  
    float d  
)
```

Parameters

a

X component of the normal defining the [Plane](#).

b

Y component of the normal defining the [Plane](#).

c

Z component of the normal defining the [Plane](#).

d

Distance of the [Plane](#) along its normal from the origin.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane Constructor (Vector3, Single)

Creates a new instance of [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane (  
    Vector3 normal,  
    float d  
)
```

Parameters

normal

The normal [vector](#) to the [Plane](#).

d

The [Plane](#)'s distance along its normal from the origin.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane Constructor (Vector3, Vector3, Vector3)

Creates a new instance of [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane (  
    Vector3 point1,  
    Vector3 point2,  
    Vector3 point3  
)
```

Parameters

point1

One point of a triangle defining the [Plane](#).

point2

One point of a triangle defining the [Plane](#).

point3

One point of a triangle defining the [Plane](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane Constructor (Vector4)

Creates a new instance of [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane (  
    Vector4 value  
)
```

Parameters

value

[Vector4](#) with X, Y, and Z components defining the normal of the [Plane](#). The W component defines the distance of the [Plane](#) along the normal from the origin.

See Also

Reference

[Plane Structure](#)














[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane Methods

Public Methods

	Name	Description
	Dot	Overloaded. Calculates the dot product of a specified Vector4 and this Plane .
	DotCoordinate	Overloaded. Returns the dot product of a specified Vector3 and the Normal vector of this Plane plus the D constant value of the Plane .
	DotNormal	Overloaded. Returns the dot product of a specified Vector3 and the Normal vector of this Plane .
	Equals	Overloaded. Determines whether two instances of Plane are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether a Plane intersects a bounding volume.
	Normalize	Overloaded. Changes the coefficients of the Normal vector of a Plane to make it of unit length.
	op_Equality	Determines whether two instances of Plane are equal.
	op_Inequality	Determines whether two instances of Plane are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current Plane .
	Transform	Overloaded. Transforms a normalized Plane by a Matrix or Quaternion .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Plane Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.Dot Method

Calculates the dot product of a specified [Vector4](#) and this [Plane](#).

Overload List

Name	Description
Plane.Dot (Vector4)	Calculates the dot product of a specified Vector4 and this Plane .
Plane.Dot (Vector4, Single)	Calculates the dot product of a specified Vector4 and this Plane .

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.Dot Method (Vector4)

Calculates the dot product of a specified [Vector4](#) and this [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Dot (  
    Vector4 value  
)
```

Parameters

value

The [Vector4](#) to multiply this [Plane](#) by.

Return Value

The dot product of the specified [Vector4](#) and this [Plane](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Dot Method (Vector4, Single)

Calculates the dot product of a specified [Vector4](#) and this [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dot (  
    ref Vector4 value,  
    out float result  
)
```

Parameters

value

The [Vector4](#) to multiply this [Plane](#) by.

result

[[OutAttribute](#)] The dot product of the specified [Vector4](#) and this [Plane](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.DotCoordinate Method

Returns the dot product of a specified [Vector3](#) and the [Normal](#) vector of this [Plane](#) plus the [D](#) constant value of the `Plane`.

Overload List

Name	Description
Plane.DotCoordinate (Vector3)	Returns the dot product of a specified Vector3 and the Normal vector of this Plane plus the distance (D) value of the <code>Plane</code> .
Plane.DotCoordinate (Vector3, Single)	Returns the dot product of a specified Vector3 and the Normal vector of this Plane plus the distance (D) value of the <code>Plane</code> .

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.DotCoordinate Method (Vector3)

Returns the dot product of a specified [Vector3](#) and the [Normal](#) vector of this [Plane](#) plus the distance ([D](#)) value of the [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float DotCoordinate (  
    Vector3 value  
)
```

Parameters

value

The [Vector3](#) to multiply by.

Return Value

The resulting value.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.DotCoordinate Method (Vector3, Single)

Returns the dot product of a specified [Vector3](#) and the [Normal](#) vector of this [Plane](#) plus the distance (D) value of the [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DotCoordinate (  
    ref Vector3 value,  
    out float result  
)
```

Parameters

value

The [Vector3](#) to multiply by.

result

[[OutAttribute](#)] The resulting value.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.DotNormal Method

Returns the dot product of a specified [Vector3](#) and the [Normal](#) vector of this [Plane](#).

Overload List

Name	Description
Plane.DotNormal (Vector3)	Returns the dot product of a specified Vector3 and the Normal vector of this Plane .
Plane.DotNormal (Vector3, Single)	Returns the dot product of a specified Vector3 and the Normal vector of this Plane .

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.DotNormal Method (Vector3)

Returns the dot product of a specified [Vector3](#) and the [Normal](#) vector of this [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float DotNormal (  
    Vector3 value  
)
```

Parameters

value

The [Vector3](#) to multiply by.

Return Value

The resulting dot product.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.DotNormal Method (Vector3, Single)

Returns the dot product of a specified [Vector3](#) and the [Normal](#) vector of this [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DotNormal (  
    ref Vector3 value,  
    out float result  
)
```

Parameters

value

The [Vector3](#) to multiply by.

result

[[OutAttribute](#)] The resulting dot product.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Equals Method

Determines whether two instances of [Plane](#) are equal.

Overload List

Name	Description
Plane.Equals (Object)	Determines whether the specified Object is equal to the Plane .
Plane.Equals (Plane)	Determines whether the specified Plane is equal to the Plane .
Plane.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.Equals Method (Object)

Determines whether the specified [Object](#) is equal to the [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [Plane](#).

Return Value

true if the specified [Object](#) is equal to the current [Plane](#); **false** otherwise.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Equals Method (Plane)

Determines whether the specified [Plane](#) is equal to the [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Plane other  
)
```

Parameters

other

The [Plane](#) to compare with the current [Plane](#).

Return Value

true if the specified [Plane](#) is equal to the current [Plane](#); **false** otherwise.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [Plane](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Intersects Method

Checks whether a [Plane](#) intersects a bounding volume.

Overload List

Name	Description
Plane.Intersects (BoundingBox)	Checks whether the current Plane intersects a specified BoundingBox .
Plane.Intersects (BoundingBox, PlaneIntersectionType)	Checks whether the current Plane intersects a BoundingBox .
Plane.Intersects (BoundingFrustum)	Checks whether the current Plane intersects a specified BoundingFrustum .
Plane.Intersects (BoundingSphere)	Checks whether the current Plane intersects a specified BoundingSphere .
Plane.Intersects (BoundingSphere, PlaneIntersectionType)	Checks whether the current Plane intersects a BoundingSphere .

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.Intersects Method (BoundingBox)

Checks whether the current [Plane](#) intersects a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneIntersectionType Intersects (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to test for intersection with.

Return Value

An enumeration indicating the relationship between the [Plane](#) and the [BoundingBox](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Intersects Method (BoundingBox, PlaneIntersectionType)

Checks whether the current [Plane](#) intersects a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox box,  
    out PlaneIntersectionType result  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] An enumeration indicating whether the [Plane](#) intersects the [BoundingBox](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Intersects Method (BoundingFrustum)

Checks whether the current [Plane](#) intersects a specified [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneIntersectionType Intersects (  
    BoundingFrustum frustum  
)
```

Parameters

frustum

The [BoundingFrustum](#) to check for intersection with.

Return Value

An enumeration indicating the relationship between the [Plane](#) and the [BoundingFrustum](#).

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Intersects Method (BoundingSphere)

Checks whether the current [Plane](#) intersects a specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneIntersectionType Intersects (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to check for intersection with.

Return Value

An enumeration indicating the relationship between the [Plane](#) and the [BoundingSphere](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Intersects Method (BoundingSphere, PlaneIntersectionType)

Checks whether the current [Plane](#) intersects a [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingSphere sphere,  
    out PlaneIntersectionType result  
)
```

Parameters

sphere

The [BoundingSphere](#) to check for intersection with.

result

[[OutAttribute](#)] An enumeration indicating whether the [Plane](#) intersects the [BoundingSphere](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Normalize Method

Changes the coefficients of the [Normal](#) vector of a [Plane](#) to make it of unit length.

Overload List

Name	Description
Plane.Normalize ()	Changes the coefficients of the Normal vector of this Plane to make it of unit length.
Plane.Normalize (Plane)	Changes the coefficients of the Normal vector of a Plane to make it of unit length.
Plane.Normalize (Plane, Plane)	Changes the coefficients of the Normal vector of a Plane to make it of unit length.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.Normalize Method ()

Changes the coefficients of the [Normal](#) vector of this [Plane](#) to make it of unit length.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Normalize ()
```

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Normalize Method (Plane)

Changes the coefficients of the [Normal](#) vector of a [Plane](#) to make it of unit length.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Plane Normalize (  
    Plane value  
)
```

Parameters

value

The [Plane](#) to normalize.

Return Value

A new [Plane](#) with a normal having unit length.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Normalize Method (Plane, Plane)

Changes the coefficients of the [Normal](#) vector of a [Plane](#) to make it of unit length.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Normalize (  
    ref Plane value,  
    out Plane result  
)
```

Parameters

value

The [Plane](#) to normalize.

result

[[OutAttribute](#)] An existing plane [Plane](#) filled in with a normalized version of the specified plane.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.op_Equality Method

Determines whether two instances of [Plane](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Plane lhs,  
    Plane rhs  
)
```

Parameters

lhs

The object to the left of the equality operator.

rhs

The object to the right of the equality operator.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.op_Inequality Method

Determines whether two instances of [Plane](#) are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Plane lhs,  
    Plane rhs  
)
```

Parameters

lhs

The object to the left of the inequality operator.

rhs

The object to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.ToString Method

Returns a [String](#) that represents the current [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of the current [Plane](#).

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Transform Method

Transforms a normalized [Plane](#) by a [Matrix](#) or [Quaternion](#).

Overload List

Name	Description
Plane.Transform (Plane, Matrix)	Transforms a normalized Plane by a Matrix .
Plane.Transform (Plane, Matrix, Plane)	Transforms a normalized Plane by a Matrix .
Plane.Transform (Plane, Quaternion)	Transforms a normalized Plane by a Quaternion rotation.
Plane.Transform (Plane, Quaternion, Plane)	Transforms a normalized Plane by a Quaternion rotation.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Plane.Transform Method (Plane, Matrix)

Transforms a normalized [Plane](#) by a [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Plane Transform (  
    Plane plane,  
    Matrix matrix  
)
```

Parameters

plane

The normalized [Plane](#) to transform. This `Plane` must already be normalized, so that its [Normal](#) vector is of unit length, before this method is called.

matrix

The transform [Matrix](#) to apply to the [Plane](#).

Return Value

A new [Plane](#) that results from applying the transform.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Transform Method (Plane, Matrix, Plane)

Transforms a normalized [Plane](#) by a [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Plane plane,  
    ref Matrix matrix,  
    out Plane result  
)
```

Parameters

plane

The normalized [Plane](#) to transform. This `Plane` must already be normalized, so that its [Normal](#) vector is of unit length, before this method is called.

matrix

The transform [Matrix](#) to apply to the [Plane](#).

result

[[OutAttribute](#)] An existing [Plane](#) filled in with the results of applying the transform.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Transform Method (Plane, Quaternion)

Transforms a normalized [Plane](#) by a [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Plane Transform (  
    Plane plane,  
    Quaternion rotation  
)
```

Parameters

plane

The normalized [Plane](#) to transform. This `Plane` must already be normalized, so that its [Normal](#) vector is of unit length, before this method is called.

rotation

The [Quaternion](#) rotation to apply to the [Plane](#).

Return Value

A new [Plane](#) that results from applying the rotation.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Plane.Transform Method (Plane, Quaternion, Plane)

Transforms a normalized [Plane](#) by a [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Plane plane,  
    ref Quaternion rotation,  
    out Plane result  
)
```

Parameters

plane

The normalized [Plane](#) to transform. This `Plane` must already be normalized, so that its [Normal](#) vector is of unit length, before this method is called.

rotation

The [Quaternion](#) rotation to apply to the [Plane](#).

result

[[OutAttribute](#)] An existing [Plane](#) filled in with the results of applying the rotation.

See Also

Reference

[Plane Structure](#)

[Plane Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlaneIntersectionType Enumeration

Describes the intersection between a plane and a bounding volume.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum PlaneIntersectionType
```

Members

Member name	Description
Back	There is no intersection, and the bounding volume is in the negative half-space of the Plane .
Front	There is no intersection, and the bounding volume is in the positive half-space of the Plane .
Intersecting	The Plane is intersected.

See Also

Reference

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlayerIndex Enumeration

Specifies the game controller associated with a player.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum PlayerIndex
```

Members

Member name	Description
One	The first controller.
Two	The second controller.
Three	The third controller.
Four	The fourth controller.

See Also

Reference

[SignedInGamer.PlayerIndex Property](#)

[SignedInGamerCollection.Item Property \(PlayerIndex\)](#)

[GamePad.GetCapabilities Method](#)

[GamePad.GetState Method](#)

[GamePad.SetVibration Method](#)

[Guide.BeginShowKeyboardInput Method](#)

[Guide.BeginShowMessageBox Method](#)

[Guide.BeginShowStorageDeviceSelector Method](#)

[Guide.ShowComposeMessage Method](#)

[Guide.ShowFriendRequest Method](#)

[Guide.ShowFriends Method](#)

[Guide.ShowGameInvite Method](#)

[Guide.ShowGamerCard Method](#)

[Guide.ShowMarketplace Method](#)

[Guide.ShowMessages Method](#)

[Guide.ShowPlayerReview Method](#)

[Guide.ShowPlayers Method](#)

[Keyboard.GetState Method](#)

[StorageDevice.OpenContainer Method](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Application Model Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point Structure

Defines a point in 2D space.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

```
C#  
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.PointConverter)")]  
[SerializableAttribute]  
public struct Point : IEquatable<Point>
```

See Also

Reference

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Point Members

The following tables list the members exposed by the Point type.


Public Constructors

	Name	Description
	Point	Initializes a new instance of Point .








Public Fields

	Name	Description
	X	Specifies the x-coordinate of the Point .
	Y	Specifies the y-coordinate of the Point .



Public Properties

	Name	Description
	Zero	Returns the point (0,0).

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two Point instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two Point instances are equal.
	Op_Inequality	Determines whether two Point instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current Point .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[Point Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Point Fields

Public Fields

	Name	Description
	X	Specifies the x-coordinate of the Point .
	Y	Specifies the y-coordinate of the Point .

See Also

Reference

[Point Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Point.X Field

Specifies the x-coordinate of the [Point](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int X
```

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point.Y Field

Specifies the y-coordinate of the [Point](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Y
```

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point Constructor

Initializes a new instance of [Point](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Point (  
    int x,  
    int y  
)
```

Parameters

x

The x-coordinate of the [Point](#).

y

The y-coordinate of the [Point](#).

See Also

Reference

[Point Structure](#)








[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two Point instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two Point instances are equal.
	op_Inequality	Determines whether two Point instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current Point .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Point Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Point.Equals Method

Determines whether two [Point](#) instances are equal.

Overload List

Name	Description
Point.Equals (Object)	Determines whether two Point instances are equal.
Point.Equals (Point)	Determines whether two Point instances are equal.
Point.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Point.Equals Method (Object)

Determines whether two [Point](#) instances are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point.Equals Method (Point)

Determines whether two [Point](#) instances are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Point other  
)
```

Parameters

other

The [Point](#) to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point.op_Equality Method

Determines whether two [Point](#) instances are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Point a,  
    Point b  
)
```

Parameters

a

[Point](#) on the left side of the equal sign.

b

[Point](#) on the right side of the equal sign.

Return Value

true if *a* and *b* are equal; **false** otherwise.

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point.op_Inequality Method

Determines whether two [Point](#) instances are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Point a,  
    Point b  
)
```

Parameters

a

The [Point](#) on the left side of the equal sign.

b

The [Point](#) on the right side of the equal sign.

Return Value

true if the instances are not equal; **false** otherwise.

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point.ToString Method

Returns a [String](#) that represents the current [Point](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

A [String](#) that represents the current [Point](#).

See Also

Reference

[Point Structure](#)


[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Point Properties

Public Properties

	Name	Description
	Zero	Returns the point (0,0).

See Also

Reference

[Point Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Point.Zero Property

Returns the point (0,0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Point Zero { get; }
```

Property Value

The point (0,0).

See Also

Reference

[Point Structure](#)

[Point Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerLineStatus Enumeration

Indicates the status of the system powerline.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum PowerLineStatus
```

Members

Member name	Description
Offline	The system is offline.
Online	The system is online.
Unknown	The power status of the system is unknown.

Remarks

The [PowerStatus.PowerLineStatus](#) property returns a **PowerLineStatus** value.

See Also

Reference

[PowerStatus.PowerLineStatus Property](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerStatus Class

Contains system power information, including battery life status and power line status.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class PowerStatus
```

Example

The example below retrieves power status details.

C#

```
GraphicsDevice.Clear(Color.CornflowerBlue);
Vector2 pos = Vector2.Zero;
SpriteFont font = Content.Load<SpriteFont>("Font");
string state = "";

state += "\r\nChargeStatus:" +
    PowerStatus.BatteryChargeStatus;
state += "\r\nBatteryFullLifetime:" +
    PowerStatus.BatteryFullLifetime.ToString();
state += "\r\nBatteryLifePercent:" +
    PowerStatus.BatteryLifePercent.ToString();
state += "\r\nBatteryLifeRemaining:" +
    PowerStatus.BatteryLifeRemaining.ToString();
state += "\r\nPowerLineStatus:" +
    PowerStatus.PowerLineStatus;

spriteBatch.Begin();
spriteBatch.DrawString(font, state, pos, Color.White);
spriteBatch.End();
```

See Also

Reference

[PowerStatus Members](#)






[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






PowerStatus Members

The following tables list the members exposed by the PowerStatus type.



Public Properties

	Name	Description
	BatteryChargeStatus	Gets the BatteryChargeStatus associated with the system.
	BatteryFullLifetime	Gets the full charge lifetime of the primary battery power source.
	BatteryLifePercent	Gets the approximate percentage of full battery time remaining.
	BatteryLifeRemaining	Gets the approximate battery life remaining.
	PowerLineStatus	Gets the PowerLineStatus associated with the system.


Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Public Events

	Name	Description
	PowerStateChanged	Raised when the power state changes.

See Also






Reference

[PowerStatus Class](#)



[Microsoft.Xna.Framework Namespace](#)

PowerStatus Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[PowerStatus Class](#)

[Microsoft.Xna.Framework Namespace](#)

PowerStatus Properties

Public Properties

	Name	Description
	BatteryChargeStatus	Gets the BatteryChargeStatus associated with the system.
	BatteryFullLifetime	Gets the full charge lifetime of the primary battery power source.
	BatteryLifePercent	Gets the approximate percentage of full battery time remaining.
	BatteryLifeRemaining	Gets the approximate battery life remaining.
	PowerLineStatus	Gets the PowerLineStatus associated with the system.

See Also

Reference

[PowerStatus Class](#)

[Microsoft.Xna.Framework Namespace](#)

PowerStatus.BatteryChargeStatus Property

Gets the [BatteryChargeStatus](#) associated with the system.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static BatteryChargeStatus BatteryChargeStatus { get; }
```

Property Value

The [BatteryChargeStatus](#) associated with the system.

See Also

Reference

[PowerStatus Class](#)

[BatteryChargeStatus Enumeration](#)

[PowerStatus Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerStatus.BatteryFullLifetime Property

Gets the full charge lifetime of the primary battery power source.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Nullable<TimeSpan> BatteryFullLifetime { get; }
```

Property Value

The full charge lifetime of the primary battery power source.

Remarks Returns the reported number of seconds of battery life available when the battery is fully charged. Returns -1 if the battery life is unknown.

See Also

Reference

[PowerStatus Class](#)

[PowerStatus Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerStatus.BatteryLifePercent Property

Gets the approximate percentage of full battery time remaining.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Nullable<float> BatteryLifePercent { get; }
```

Property Value

Approximate percentage of full battery time remaining.

Remarks Returns the approximate percentage, from 0 to 100, of full battery time remaining. Returns 255 if the percentage is unknown.

See Also

Reference

[PowerStatus Class](#)

[PowerStatus Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerStatus.BatteryLifeRemaining Property

Gets the approximate battery life remaining.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Nullable<TimeSpan> BatteryLifeRemaining { get; }
```

Property Value

Approximate battery life remaining.

Remarks Returns the approximate number of seconds of battery life remaining, or -1 if the approximate remaining battery life is unknown.

See Also

Reference

[PowerStatus Class](#)

[PowerStatus Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerStatus.PowerLineStatus Property

Gets the [PowerLineStatus](#) associated with the system.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static PowerLineStatus PowerLineStatus { get; }
```

Property Value

The [PowerLineStatus](#) associated with the system.

See Also

Reference

[PowerStatus Class](#)

[PowerLineStatus Enumeration](#)


[PowerStatus Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PowerStatus Events

Public Events

	Name	Description
 S	PowerStateChanged	Raised when the power state changes.

See Also

Reference

[PowerStatus Class](#)

[Microsoft.Xna.Framework Namespace](#)

PowerStatus.PowerStateChanged Event

Raised when the power state changes.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler PowerStateChanged
```

See Also

Reference

[PowerStatus Class](#)

[PowerStatus Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PreparingDeviceSettingsEventArgs Class

Arguments for the [GraphicsDeviceManager.PreparingDeviceSettings](#) event.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public class PreparingDeviceSettingsEventArgs : EventArgs
```

See Also

Reference

[PreparingDeviceSettingsEventArgs Members](#)


[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PreparingDeviceSettingsEventArgs Members

The following tables list the members exposed by the PreparingDeviceSettingsEventArgs type.






Public Constructors

	Name	Description
	PreparingDeviceSettingsEventArgs	Creates a new instance of PreparingDeviceSettingsEventArgs.



Public Properties

	Name	Description
	GraphicsDeviceInformation	Information about the GraphicsDevice .

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PreparingDeviceSettingsEventArgs Class](#)

[Microsoft.Xna.Framework Namespace](#)

PreparingDeviceSettingsEventArgs Constructor

Creates a new instance of **PreparingDeviceSettingsEventArgs**.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public PreparingDeviceSettingsEventArgs (  
    GraphicsDeviceInformation graphicsDeviceInformation  
)
```

Parameters

graphicsDeviceInformation

Information about the [GraphicsDevice](#).

See Also

Reference

[PreparingDeviceSettingsEventArgs Class](#)






[PreparingDeviceSettingsEventArgs Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PreparingDeviceSettingsEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[PreparingDeviceSettingsEventArgs Class](#)

[Microsoft.Xna.Framework Namespace](#)

PreparingDeviceSettingsEventArgs Properties

Public Properties

	Name	Description
	GraphicsDeviceInformation	Information about the GraphicsDevice .

See Also

Reference

[PreparingDeviceSettingsEventArgs Class](#)

[Microsoft.Xna.Framework Namespace](#)

PreparingDeviceSettingsEventArgs.GraphicsDeviceInformation Property

Information about the [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework.Game (in microsoft.xna.framework.game.dll)

Syntax

C#

```
public GraphicsDeviceInformation GraphicsDeviceInformation { get; }
```

Property Value

Information about the [GraphicsDevice](#).

See Also

Reference

[PreparingDeviceSettingsEventArgs Class](#)

[PreparingDeviceSettingsEventArgs Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion Structure

Defines a four-dimensional vector (x,y,z,w), which is used to efficiently rotate an object about the (x, y, z) vector by the angle theta, where $w = \cos(\text{theta}/2)$.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.QuaternionConverter)")]
[SerializableAttribute]
public struct Quaternion : IEquatable<Quaternion>
```

Remarks Quaternions represent a rotation. Typically, they are used for smooth interpolation between two angles, and for avoiding the gimbal lock problem that can occur with euler angles.

See Also

Reference

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





Quaternion Members

The following tables list the members exposed by the Quaternion type.


Public Constructors

	Name	Description
	Quaternion	Overloaded. Initializes a new instance of Quaternion .
























Public Fields







	Name	Description
	W	Specifies the rotation component of the quaternion.
	X	Specifies the x-value of the vector component of the quaternion.
	Y	Specifies the y-value of the vector component of the quaternion.
	Z	Specifies the z-value of the vector component of the quaternion.

Public Properties



	Name	Description
	Identity	Returns a Quaternion representing no rotation.

Public Methods

	Name	Description
	Add	Overloaded. Adds two Quaternions .
	Concatenate	Overloaded. Concatenates two Quaternions ; the result represents the first rotation followed by the second rotation.
	Conjugate	Overloaded. Calculates the conjugate of a Quaternion .
	CreateFromAxisAngle	Overloaded. Creates a Quaternion from a vector and an angle to rotate about the vector.
	CreateFromRotationMatrix	Overloaded. Creates a Quaternion from a rotation Matrix .
	CreateFromYawPitchRoll	Overloaded. Creates a new Quaternion from specified yaw, pitch, and roll angles.
	Divide	Overloaded. Divides a Quaternion by another Quaternion .
	Dot	Overloaded. Calculates the dot product of two Quaternions .
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Get the hash code of this object.
	GetType	(Inherited from Object .)
	Inverse	Overloaded. Returns the inverse of a Quaternion .
	Length	Calculates the length of a Quaternion .
	LengthSquared	Calculates the length squared of a Quaternion .
	Lerp	Overloaded. Linearly interpolates between two quaternions.
	Multiply	Overloaded. Multiplies a quaternion by a scalar or another quaternion.
	Negate	Overloaded. Flips the sign of each component of the quaternion.
	Normalize	Overloaded. Divides each component of a quaternion by the length of the quaternion.
	op_Addition	Adds two Quaternions .
	op_Division	Divides a Quaternion by another Quaternion .
	op_Equality	Compares two Quaternions for equality.
	op_Inequality	Compare two Quaternions for inequality.
	op_Multiply	Overloaded. Multiplies a quaternion by a scalar or another quaternion.

 op_Subtraction	Subtracts a quaternion from another quaternion.
 op_UnaryNegation	Flips the sign of each component of the quaternion.
 ReferenceEquals	(Inherited from Object .)
 Slerp	Overloaded. Interpolates between two quaternions, using spherical linear interpolation.
 Subtract	Overloaded. Subtracts a quaternion from another quaternion.
 ToString	Retireves a string representation of the current object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[Quaternion Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion Fields

Public Fields

	Name	Description
	W	Specifies the rotation component of the quaternion.
	X	Specifies the x-value of the vector component of the quaternion.
	Y	Specifies the y-value of the vector component of the quaternion.
	Z	Specifies the z-value of the vector component of the quaternion.

See Also

Reference

[Quaternion Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.W Field

Specifies the rotation component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float W
```

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.X Field

Specifies the x-value of the vector component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float X
```

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Y Field

Specifies the y-value of the vector component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Y
```

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Z Field

Specifies the z-value of the vector component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Z
```

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion Constructor

Initializes a new instance of [Quaternion](#).

Overload List

Name	Description
Quaternion (Single, Single, Single, Single)	Initializes a new instance of Quaternion .
Quaternion (Vector3, Single)	Initializes a new instance of Quaternion .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion Constructor (Single, Single, Single, Single)

Initializes a new instance of [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x
The x-value of the quaternion.

y
The y-value of the quaternion.

z
The z-value of the quaternion.

w
The w-value of the quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion Constructor (Vector3, Single)

Initializes a new instance of [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion (
    Vector3 vectorPart,
    float scalarPart
)
```

Parameters

vectorPart

The vector component of the quaternion.

scalarPart

The rotation component of the quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion Methods

Public Methods

	Name	Description
	Add	Overloaded. Adds two Quaternions .
	Concatenate	Overloaded. Concatenates two Quaternions ; the result represents the first rotation followed by the second rotation.
	Conjugate	Overloaded. Calculates the conjugate of a Quaternion .
	CreateFromAxisAngle	Overloaded. Creates a Quaternion from a vector and an angle to rotate about the vector.
	CreateFromRotationMatrix	Overloaded. Creates a Quaternion from a rotation Matrix .
	CreateFromYawPitchRoll	Overloaded. Creates a new Quaternion from specified yaw, pitch, and roll angles.
	Divide	Overloaded. Divides a Quaternion by another Quaternion .
	Dot	Overloaded. Calculates the dot product of two Quaternions .
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Get the hash code of this object.
	GetType	(Inherited from Object .)
	Inverse	Overloaded. Returns the inverse of a Quaternion .
	Length	Calculates the length of a Quaternion .
	LengthSquared	Calculates the length squared of a Quaternion .
	Lerp	Overloaded. Linearly interpolates between two quaternions.
	Multiply	Overloaded. Multiplies a quaternion by a scalar or another quaternion.
	Negate	Overloaded. Flips the sign of each component of the quaternion.
	Normalize	Overloaded. Divides each component of a quaternion by the length of the quaternion.
	op_Addition	Adds two Quaternions .
	op_Division	Divides a Quaternion by another Quaternion .
	op_Equality	Compares two Quaternions for equality.
	op_Inequality	Compare two Quaternions for inequality.
	op_Multiply	Overloaded. Multiplies a quaternion by a scalar or another quaternion.
	op_Subtraction	Subtracts a quaternion from another quaternion.
	op_UnaryNegation	Flips the sign of each component of the quaternion.
	ReferenceEquals	(Inherited from Object .)
	Slerp	Overloaded. Interpolates between two quaternions, using spherical linear interpolation.
	Subtract	Overloaded. Subtracts a quaternion from another quaternion.
	ToString	Retrieves a string representation of the current object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

Quaternion Structure
Microsoft.Xna.Framework Namespace

Quaternion.Add Method

Adds two [Quaternions](#).

Overload List

Name	Description
Quaternion.Add (Quaternion, Quaternion)	Adds two Quaternions .
Quaternion.Add (Quaternion, Quaternion, Quaternion)	Adds two Quaternions .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Add Method (Quaternion, Quaternion)

Adds two [Quaternions](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Add (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

[Quaternion](#) to add.

quaternion2

[Quaternion](#) to add.

Return Value

Result of adding the [Quaternions](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Add Method (Quaternion, Quaternion, Quaternion)

Adds two [Quaternions](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Add (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    out Quaternion result  
)
```

Parameters

quaternion1

[Quaternion](#) to add.

quaternion2

[Quaternion](#) to add.

result

[[OutAttribute](#)] Result of adding the [Quaternions](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Concatenate Method

Concatenates two [Quaternions](#); the result represents the first rotation followed by the second rotation.

Overload List

Name	Description
Quaternion.Concatenate (Quaternion, Quaternion)	Concatenates two Quaternions ; the result represents the <i>value1</i> rotation followed by the <i>value2</i> rotation.
Quaternion.Concatenate (Quaternion, Quaternion, Quaternion)	Concatenates two Quaternions ; the result represents the <i>value1</i> rotation followed by the <i>value2</i> rotation.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Concatenate Method (Quaternion, Quaternion)

Concatenates two [Quaternions](#); the result represents the *value1* rotation followed by the *value2* rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Concatenate (  
    Quaternion value1,  
    Quaternion value2  
)
```

Parameters

value1

The first [Quaternion](#) rotation in the series.

value2

The second [Quaternion](#) rotation in the series.

Return Value

A new [Quaternion](#) representing the concatenation of the *value1* rotation followed by the *value2* rotation.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Concatenate Method (Quaternion, Quaternion, Quaternion)

Concatenates two [Quaternions](#); the result represents the *value1* rotation followed by the *value2* rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Concatenate (  
    ref Quaternion value1,  
    ref Quaternion value2,  
    out Quaternion result  
)
```

Parameters

value1

The first [Quaternion](#) rotation in the series.

value2

The second [Quaternion](#) rotation in the series.

result

[[OutAttribute](#)] The [Quaternion](#) rotation representing the concatenation of *value1* followed by *value2*.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Conjugate Method

Calculates the conjugate of a [Quaternion](#).

Overload List

Name	Description
Quaternion.Conjugate ()	Transforms this Quaternion into its conjugate.
Quaternion.Conjugate (Quaternion)	Returns the conjugate of a specified Quaternion .
Quaternion.Conjugate (Quaternion, Quaternion)	Returns the conjugate of a specified Quaternion .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Conjugate Method ()

Transforms this [Quaternion](#) into its conjugate.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Conjugate ()
```

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Conjugate Method (Quaternion)

Returns the conjugate of a specified [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Conjugate (  
    Quaternion value  
)
```

Parameters

value

The [Quaternion](#) of which to return the conjugate.

Return Value

A new [Quaternion](#) that is the conjugate of the specified one.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Conjugate Method (Quaternion, Quaternion)

Returns the conjugate of a specified [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Conjugate (  
    ref Quaternion value,  
    out Quaternion result  
)
```

Parameters

value

The [Quaternion](#) of which to return the conjugate.

result

[[OutAttribute](#)] An existing [Quaternion](#) filled in to be the conjugate of the specified one.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.CreateFromAxisAngle Method

Creates a [Quaternion](#) from a vector and an angle to rotate about the vector.

Overload List

Name	Description
Quaternion.CreateFromAxisAngle (Vector3, Single)	Creates a Quaternion from a vector and an angle to rotate about the vector.
Quaternion.CreateFromAxisAngle (Vector3, Single, Quaternion)	Creates a Quaternion from a vector and an angle to rotate about the vector.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.CreateFromAxisAngle Method (Vector3, Single)

Creates a [Quaternion](#) from a vector and an angle to rotate about the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion CreateFromAxisAngle (  
    Vector3 axis,  
    float angle  
)
```

Parameters

axis

The vector to rotate around.

angle

The angle to rotate around the vector.

Return Value

The created [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.CreateFromAxisAngle Method (Vector3, Single, Quaternion)

Creates a [Quaternion](#) from a vector and an angle to rotate about the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromAxisAngle (  
    ref Vector3 axis,  
    float angle,  
    out Quaternion result  
)
```

Parameters

axis

The vector to rotate around.

angle

The angle to rotate around the vector.

result

[[OutAttribute](#)] The created [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.CreateFromRotationMatrix Method

Creates a [Quaternion](#) from a rotation [Matrix](#).

Overload List

Name	Description
Quaternion.CreateFromRotationMatrix (Matrix)	Creates a Quaternion from a rotation Matrix .
Quaternion.CreateFromRotationMatrix (Matrix, Quaternion)	Creates a Quaternion from a rotation Matrix .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.CreateFromRotationMatrix Method (Matrix)

Creates a [Quaternion](#) from a rotation [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion CreateFromRotationMatrix (  
    Matrix matrix  
)
```

Parameters

matrix

The rotation [Matrix](#) to create the [Quaternion](#) from.

Return Value

The created [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.CreateFromRotationMatrix Method (Matrix, Quaternion)

Creates a [Quaternion](#) from a rotation [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromRotationMatrix (  
    ref Matrix matrix,  
    out Quaternion result  
)
```

Parameters

matrix

The rotation [Matrix](#) to create the [Quaternion](#) from.

result

[[OutAttribute](#)] The created [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.CreateFromYawPitchRoll Method

Creates a new [Quaternion](#) from specified yaw, pitch, and roll angles.

Overload List

Name	Description
Quaternion.CreateFromYawPitchRoll (Single, Single, Single)	Creates a new Quaternion from specified yaw, pitch, and roll angles.
Quaternion.CreateFromYawPitchRoll (Single, Single, Single, Quaternion)	Creates a new Quaternion from specified yaw, pitch, and roll angles.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.CreateFromYawPitchRoll Method (Single, Single, Single)

Creates a new [Quaternion](#) from specified yaw, pitch, and roll angles.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion CreateFromYawPitchRoll (  
    float yaw,  
    float pitch,  
    float roll  
)
```

Parameters

yaw

The yaw angle, in radians, around the y-axis.

pitch

The pitch angle, in radians, around the x-axis.

roll

The roll angle, in radians, around the z-axis.

Return Value

A new [Quaternion](#) expressing the specified yaw, pitch, and roll angles.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.CreateFromYawPitchRoll Method (Single, Single, Single, Quaternion)

Creates a new [Quaternion](#) from specified yaw, pitch, and roll angles.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CreateFromYawPitchRoll (  
    float yaw,  
    float pitch,  
    float roll,  
    out Quaternion result  
)
```

Parameters

yaw

The yaw angle, in radians, around the y-axis.

pitch

The pitch angle, in radians, around the x-axis.

roll

The roll angle, in radians, around the z-axis.

result

[[OutAttribute](#)] An existing [Quaternion](#) filled in to express the specified yaw, pitch, and roll angles.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Divide Method

Divides a [Quaternion](#) by another [Quaternion](#).

Overload List

Name	Description
Quaternion.Divide (Quaternion, Quaternion)	Divides a Quaternion by another Quaternion .
Quaternion.Divide (Quaternion, Quaternion, Quaternion)	Divides a Quaternion by another Quaternion .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Divide Method (Quaternion, Quaternion)

Divides a [Quaternion](#) by another [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Divide (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

The divisor.

Return Value

Result of the division.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Divide Method (Quaternion, Quaternion, Quaternion)

Divides a [Quaternion](#) by another [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    out Quaternion result  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

The divisor.

result

[[OutAttribute](#)] Result of the division.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Dot Method

Calculates the dot product of two [Quaternions](#).

Overload List

Name	Description
Quaternion.Dot (Quaternion, Quaternion)	Calculates the dot product of two Quaternions .
Quaternion.Dot (Quaternion, Quaternion, Single)	Calculates the dot product of two Quaternions .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Dot Method (Quaternion, Quaternion)

Calculates the dot product of two [Quaternions](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Dot (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

Source [Quaternion](#).

Return Value

Dot product of the [Quaternions](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Dot Method (Quaternion, Quaternion, Single)

Calculates the dot product of two [Quaternions](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Dot (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    out float result  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

Source [Quaternion](#).

result

[[OutAttribute](#)] Dot product of the [Quaternions](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Quaternion.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Quaternion.Equals (Quaternion)	Determines whether the specified Object is equal to the Quaternion .
Quaternion.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object to make the comparison with.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Equals Method (Quaternion)

Determines whether the specified [Object](#) is equal to the [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Quaternion other  
)
```

Parameters

other

The [Quaternion](#) to compare with the current [Quaternion](#).

Return Value

true if the specified [Object](#) is equal to the current [Quaternion](#); **false** otherwise.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.GetHashCode Method

Get the hash code of this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Inverse Method

Returns the inverse of a [Quaternion](#).

Overload List

Name	Description
Quaternion.Inverse (Quaternion)	Returns the inverse of a Quaternion .
Quaternion.Inverse (Quaternion, Quaternion)	Returns the inverse of a Quaternion .

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Inverse Method (Quaternion)

Returns the inverse of a [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Inverse (  
    Quaternion quaternion  
)
```

Parameters

quaternion

Source [Quaternion](#).

Return Value

The inverse of the [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Inverse Method (Quaternion, Quaternion)

Returns the inverse of a [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Inverse (  
    ref Quaternion quaternion,  
    out Quaternion result  
)
```

Parameters

quaternion

Source [Quaternion](#).

result

[[OutAttribute](#)] The inverse of the [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Length Method

Calculates the length of a [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Length ()
```

Return Value

The length of the [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.LengthSquared Method

Calculates the length squared of a [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float LengthSquared ()
```

Return Value

The length squared of the [Quaternion](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Lerp Method

Linearly interpolates between two quaternions.

Overload List

Name	Description
Quaternion.Lerp (Quaternion, Quaternion, Single)	Linearly interpolates between two quaternions.
Quaternion.Lerp (Quaternion, Quaternion, Single, Quaternion)	Linearly interpolates between two quaternions.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Lerp Method (Quaternion, Quaternion, Single)

Linearly interpolates between two quaternions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Lerp (  
    Quaternion quaternion1,  
    Quaternion quaternion2,  
    float amount  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

amount

Value indicating how far to interpolate between the quaternions.

Return Value

The resulting quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Lerp Method (Quaternion, Quaternion, Single, Quaternion)

Linearly interpolates between two quaternions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Lerp (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    float amount,  
    out Quaternion result  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

amount

Value indicating how far to interpolate between the quaternions.

result

[[OutAttribute](#)] The resulting quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Multiply Method

Multiplies a quaternion by a scalar or another quaternion.

Overload List

Name	Description
Quaternion.Multiply (Quaternion, Quaternion)	Multiplies two quaternions.
Quaternion.Multiply (Quaternion, Quaternion, Quaternion)	Multiplies two quaternions.
Quaternion.Multiply (Quaternion, Single)	Multiplies a quaternion by a scalar value.
Quaternion.Multiply (Quaternion, Single, Quaternion)	Multiplies a quaternion by a scalar value.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Multiply Method (Quaternion, Quaternion)

Multiplies two quaternions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Multiply (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

The quaternion on the left of the multiplication.

quaternion2

The quaternion on the right of the multiplication.

Return Value

The result of the multiplication.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Multiply Method (Quaternion, Quaternion, Quaternion)

Multiplies two quaternions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    out Quaternion result  
)
```

Parameters

quaternion1

The quaternion on the left of the multiplication.

quaternion2

The quaternion on the right of the multiplication.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Multiply Method (Quaternion, Single)

Multiplies a quaternion by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Multiply (  
    Quaternion quaternion1,  
    float scaleFactor  
)
```

Parameters

quaternion1

Source quaternion.

scaleFactor

Scalar value.

Return Value

The result of the multiplication.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Multiply Method (Quaternion, Single, Quaternion)

Multiplies a quaternion by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Quaternion quaternion1,  
    float scaleFactor,  
    out Quaternion result  
)
```

Parameters

quaternion1

Source quaternion.

scaleFactor

Scalar value.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Negate Method

Flips the sign of each component of the quaternion.

Overload List

Name	Description
Quaternion.Negate (Quaternion)	Flips the sign of each component of the quaternion.
Quaternion.Negate (Quaternion, Quaternion)	Flips the sign of each component of the quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Negate Method (Quaternion)

Flips the sign of each component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Negate (  
    Quaternion quaternion  
)
```

Parameters

quaternion

Source quaternion.

Return Value

Negated quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Negate Method (Quaternion, Quaternion)

Flips the sign of each component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Negate (  
    ref Quaternion quaternion,  
    out Quaternion result  
)
```

Parameters

quaternion

Source quaternion.

result

[[OutAttribute](#)] Negated quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Normalize Method

Divides each component of a quaternion by the length of the quaternion.

Overload List

Name	Description
Quaternion.Normalize ()	Divides each component of the quaternion by the length of the quaternion.
Quaternion.Normalize (Quaternion)	Divides each component of the quaternion by the length of the quaternion.
Quaternion.Normalize (Quaternion, Quaternion)	Divides each component of the quaternion by the length of the quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Normalize Method ()

Divides each component of the quaternion by the length of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Normalize ()
```

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Normalize Method (Quaternion)

Divides each component of the quaternion by the length of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Normalize (  
    Quaternion quaternion  
)
```

Parameters

quaternion

Source quaternion.

Return Value

Normalized quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Normalize Method (Quaternion, Quaternion)

Divides each component of the quaternion by the length of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Normalize (  
    ref Quaternion quaternion,  
    out Quaternion result  
)
```

Parameters

quaternion

Source quaternion.

result

[[OutAttribute](#)] Normalized quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Addition Method

Adds two [Quaternions](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion op_Addition (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

[Quaternion](#) to add.

quaternion2

[Quaternion](#) to add.

Return Value

Result of adding the [Quaternions](#).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Division Method

Divides a [Quaternion](#) by another [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion op_Division (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

The divisor.

Return Value

Result of the division.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Equality Method

Compares two [Quaternions](#) for equality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

Source [Quaternion](#).

Return Value

true if the [Quaternions](#) are equal, **false** otherwise.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Inequality Method

Compare two [Quaternions](#) for inequality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source [Quaternion](#).

quaternion2

Source [Quaternion](#).

Return Value

true if the [Quaternions](#) are not equal; **false** otherwise.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Multiply Method

Multiplies a quaternion by a scalar or another quaternion.

Overload List

Name	Description
Quaternion.op_Multiply (Quaternion, Quaternion)	Multiplies two quaternions.
Quaternion.op_Multiply (Quaternion, Single)	Multiplies a quaternion by a scalar value.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.op_Multiply Method (Quaternion, Quaternion)

Multiplies two quaternions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion op_Multiply (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

Return Value

The result of the multiplication.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Multiply Method (Quaternion, Single)

Multiplies a quaternion by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion op_Multiply (  
    Quaternion quaternion1,  
    float scaleFactor  
)
```

Parameters

quaternion1

Source quaternion.

scaleFactor

Scalar value.

Return Value

The result of the multiplication.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_Subtraction Method

Subtracts a quaternion from another quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion op_Subtraction (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

Return Value

Result of the subtraction.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.op_UnaryNegation Method

Flips the sign of each component of the quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion op_UnaryNegation (  
    Quaternion quaternion  
)
```

Parameters

quaternion

Source quaternion.

Return Value

Negated quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Slerp Method

Interpolates between two quaternions, using spherical linear interpolation.

Overload List

Name	Description
Quaternion.Slerp (Quaternion, Quaternion, Single)	Interpolates between two quaternions, using spherical linear interpolation.
Quaternion.Slerp (Quaternion, Quaternion, Single, Quaternion)	Interpolates between two quaternions, using spherical linear interpolation.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Slerp Method (Quaternion, Quaternion, Single)

Interpolates between two quaternions, using spherical linear interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Slerp (  
    Quaternion quaternion1,  
    Quaternion quaternion2,  
    float amount  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

amount

Value that indicates how far to interpolate between the quaternions.

Return Value

Result of the interpolation.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Slerp Method (Quaternion, Quaternion, Single, Quaternion)

Interpolates between two quaternions, using spherical linear interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Slerp (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    float amount,  
    out Quaternion result  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

amount

Value that indicates how far to interpolate between the quaternions.

result

[[OutAttribute](#)] Result of the interpolation.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Subtract Method

Subtracts a quaternion from another quaternion.

Overload List

Name	Description
Quaternion.Subtract (Quaternion, Quaternion)	Subtracts a quaternion from another quaternion.
Quaternion.Subtract (Quaternion, Quaternion, Quaternion)	Subtracts a quaternion from another quaternion.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Subtract Method (Quaternion, Quaternion)

Subtracts a quaternion from another quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Subtract (  
    Quaternion quaternion1,  
    Quaternion quaternion2  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

Return Value

Result of the subtraction.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.Subtract Method (Quaternion, Quaternion, Quaternion)

Subtracts a quaternion from another quaternion.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Subtract (  
    ref Quaternion quaternion1,  
    ref Quaternion quaternion2,  
    out Quaternion result  
)
```

Parameters

quaternion1

Source quaternion.

quaternion2

Source quaternion.

result

[[OutAttribute](#)] Result of the subtraction.

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion.ToString Method

Retrieves a string representation of the current object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Quaternion Structure](#)


[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Quaternion Properties

Public Properties

	Name	Description
	S Identity	Returns a Quaternion representing no rotation.

See Also

Reference

[Quaternion Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Quaternion.Identity Property

Returns a [Quaternion](#) representing no rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Quaternion Identity { get; }
```

Property Value

The identity [Quaternion](#) (0,0,0,1).

See Also

Reference

[Quaternion Structure](#)

[Quaternion Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray Structure

Defines a ray.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.RayConverter)")]  
[SerializableAttribute()]  
public struct Ray : IEquatable<Ray>
```

See Also

Reference

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[Math Overview](#)


[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Ray Members

The following tables list the members exposed by the Ray type.









Public Constructors

	Name	Description
	Ray	Creates a new instance of Ray .



Public Fields

	Name	Description
	Direction	Unit vector specifying the direction the Ray is pointing.
	Position	Specifies the starting point of the Ray .

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of Ray are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the Ray intersects a specified plane or bounding volume.
	Op_Equality	Determines whether two instances of Ray are equal.
	Op_Inequality	Determines whether two instances of Ray are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current Ray .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[Ray Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Ray Fields

Public Fields

	Name	Description
	Direction	Unit vector specifying the direction the Ray is pointing.
	Position	Specifies the starting point of the Ray .

See Also

Reference

[Ray Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Ray.Direction Field

Unit vector specifying the direction the [Ray](#) is pointing.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Direction
```

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Position Field

Specifies the starting point of the [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position
```

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray Constructor

Creates a new instance of [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Ray (  
    Vector3 position,  
    Vector3 direction  
)
```

Parameters

position

The starting point of the [Ray](#).

direction

Unit vector describing the direction of the [Ray](#).

See Also

Reference

[Ray Structure](#)









[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of Ray are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Intersects	Overloaded. Checks whether the Ray intersects a specified plane or bounding volume.
	Op_Equality	Determines whether two instances of Ray are equal.
	Op_Inequality	Determines whether two instances of Ray are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current Ray .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Ray Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Ray.Equals Method

Determines whether two instances of [Ray](#) are equal.

Overload List

Name	Description
Ray.Equals (Object)	Determines whether two instances of Ray are equal.
Ray.Equals (Ray)	Determines whether the specified Ray is equal to the current Ray .
Ray.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Ray.Equals Method (Object)

Determines whether two instances of [Ray](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [Ray](#).

Return Value

true if the specified [Object](#) is equal to the current [Ray](#); **false** otherwise.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Equals Method (Ray)

Determines whether the specified [Ray](#) is equal to the current [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Ray other  
)
```

Parameters

other

The [Ray](#) to compare with the current [Ray](#).

Return Value

true if the specified [Ray](#) is equal to the current [Ray](#); **false** otherwise.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [Ray](#).

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method

Checks whether the [Ray](#) intersects a specified plane or bounding volume.

Overload List

Name	Description
Ray.Intersects (BoundingBox)	Checks whether the Ray intersects a specified BoundingBox .
Ray.Intersects (BoundingBox, Nullable<Single>)	Checks whether the current Ray intersects a BoundingBox .
Ray.Intersects (BoundingFrustum)	Checks whether the Ray intersects a specified BoundingFrustum .
Ray.Intersects (BoundingSphere)	Checks whether the Ray intersects a specified BoundingSphere .
Ray.Intersects (BoundingSphere, Nullable<Single>)	Checks whether the current Ray intersects a BoundingSphere .
Ray.Intersects (Plane)	Determines whether this Ray intersects a specified Plane .
Ray.Intersects (Plane, Nullable<Single>)	Determines whether this Ray intersects a specified Plane .

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Detect Whether a User Clicked a 3D Object](#)

Ray.Intersects Method (BoundingBox)

Checks whether the [Ray](#) intersects a specified [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    BoundingBox box  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with the [Ray](#).

Return Value

Distance at which the ray intersects the [BoundingBox](#) or **null** if there is no intersection.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method (BoundingBox, Nullable<Single>)

Checks whether the current [Ray](#) intersects a [BoundingBox](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingBox box,  
    out Nullable<float> result  
)
```

Parameters

box

The [BoundingBox](#) to check for intersection with.

result

[[OutAttribute](#)] Distance at which the ray intersects the [BoundingBox](#) or **null** if there is no intersection.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method (BoundingFrustum)

Checks whether the [Ray](#) intersects a specified [BoundingFrustum](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    BoundingFrustum frustum  
)
```

Parameters

frustum

The [BoundingFrustum](#) to check for intersection with the [Ray](#).

Return Value

Distance at which the ray intersects the [BoundingFrustum](#) or **null** if there is no intersection.

Exceptions

Exception type	Condition
ArgumentNullException	<i>frustum</i> is null .

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method (BoundingSphere)

Checks whether the [Ray](#) intersects a specified [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    BoundingSphere sphere  
)
```

Parameters

sphere

The [BoundingSphere](#) to check for intersection with the [Ray](#).

Return Value

Distance at which the ray intersects the [BoundingSphere](#) or **null** if there is no intersection.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method (BoundingSphere, Nullable<Single>)

Checks whether the current [Ray](#) intersects a [BoundingSphere](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref BoundingSphere sphere,  
    out Nullable<float> result  
)
```

Parameters

sphere

The [BoundingSphere](#) to check for intersection with.

result

[[OutAttribute](#)] Distance at which the ray intersects the [BoundingSphere](#) or **null** if there is no intersection.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method (Plane)

Determines whether this [Ray](#) intersects a specified [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<float> Intersects (  
    Plane plane  
)
```

Parameters

plane

The [Plane](#) with which to calculate this [Ray](#)'s intersection.

Return Value

The distance at which this [Ray](#) intersects the specified [Plane](#), or **null** if there is no intersection.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.Intersects Method (Plane, Nullable<Single>)

Determines whether this [Ray](#) intersects a specified [Plane](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Plane plane,  
    out Nullable<float> result  
)
```

Parameters

plane

The [Plane](#) with which to calculate this [Ray](#)'s intersection.

result

[[OutAttribute](#)] The distance at which this [Ray](#) intersects the specified [Plane](#), or **null** if there is no intersection.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.op_Equality Method

Determines whether two instances of [Ray](#) are equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Ray a,  
    Ray b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.op_Inequality Method

Determines whether two instances of [Ray](#) are not equal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Ray a,  
    Ray b  
)
```

Parameters

a

The object to the left of the inequality operator.

b

The object to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Ray.ToString Method

Returns a [String](#) that represents the current [Ray](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of the current [Ray](#).

See Also

Reference

[Ray Structure](#)

[Ray Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle Structure

Defines a rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.RectangleConverter)")]  
[SerializableAttribute]  
public struct Rectangle : IEquatable<Rectangle>
```

See Also

Reference

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





Rectangle Members

The following tables list the members exposed by the Rectangle type.









Public Constructors

	Name	Description
	Rectangle	Initializes a new instance of Rectangle .














Public Fields

	Name	Description
	Height	Specifies the height of the rectangle.
	Width	Specifies the width of the rectangle.
	X	Specifies the x-coordinate of the rectangle.
	Y	Specifies the y-coordinate of the rectangle.



Public Properties

	Name	Description
	Bottom	Returns the y-coordinate of the bottom of the rectangle.
	Center	Gets the Point that specifies the center of the rectangle.
	Empty	Returns a Rectangle with all of its values set to zero.
	IsEmpty	Gets a value that indicates whether the Rectangle is empty.
	Left	Returns the x-coordinate of the left side of the rectangle.
	Location	Gets or sets the upper-left value of the Rectangle .
	Right	Returns the x-coordinate of the right side of the rectangle.
	Top	Returns the y-coordinate of the top of the rectangle.

Public Methods

	Name	Description
	Contains	Overloaded. Determines whether this Rectangle contains a specified point or Rectangle .
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Inflate	Pushes the edges of the Rectangle out by the horizontal and vertical values specified.
	Intersect	Overloaded. Creates a Rectangle defining the area where one rectangle overlaps another rectangle.
	Intersects	Overloaded. Determines whether a specified Rectangle intersects with this Rectangle .
	Offset	Overloaded. Changes the position of the Rectangle .
	Op_Equality	Compares two rectangles for equality.
	Op_Inequality	Compares two rectangles for inequality.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of the current object.
	Union	Overloaded. Creates a new Rectangle that exactly contains two other rectangles.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[Rectangle Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle Fields

Public Fields

	Name	Description
	Height	Specifies the height of the rectangle.
	Width	Specifies the width of the rectangle.
	X	Specifies the x-coordinate of the rectangle.
	Y	Specifies the y-coordinate of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Height Field

Specifies the height of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height
```

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Width Field

Specifies the width of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width
```

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.X Field

Specifies the x-coordinate of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int X
```

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Y Field

Specifies the y-coordinate of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Y
```

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle Constructor

Initializes a new instance of [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rectangle (  
    int x,  
    int y,  
    int width,  
    int height  
)
```

Parameters

x

The x-coordinate of the rectangle.

y

The y-coordinate of the rectangle.

width

Width of the rectangle.

height

Height of the rectangle.

See Also

Reference

[Rectangle Structure](#)


















[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle Methods

Public Methods

	Name	Description
	Contains	Overloaded. Determines whether this Rectangle contains a specified point or Rectangle .
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Inflate	Pushes the edges of the Rectangle out by the horizontal and vertical values specified.
 	Intersect	Overloaded. Creates a Rectangle defining the area where one rectangle overlaps another rectangle.
	Intersects	Overloaded. Determines whether a specified Rectangle intersects with this Rectangle .
	Offset	Overloaded. Changes the position of the Rectangle .
 	Op_Equality	Compares two rectangles for equality.
 	Op_Inequality	Compares two rectangles for inequality.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of the current object.
 	Union	Overloaded. Creates a new Rectangle that exactly contains two other rectangles.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Rectangle Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Contains Method

Determines whether this [Rectangle](#) contains a specified point or [Rectangle](#).

Overload List

Name	Description
Rectangle.Contains (Int32, Int32)	Determines whether this Rectangle contains a specified point represented by its x- and y-coordinates.
Rectangle.Contains (Point)	Determines whether this Rectangle contains a specified Point .
Rectangle.Contains (Point, Boolean)	Determines whether this Rectangle contains a specified Point .
Rectangle.Contains (Rectangle)	Determines whether this Rectangle entirely contains a specified Rectangle .
Rectangle.Contains (Rectangle, Boolean)	Determines whether this Rectangle entirely contains a specified Rectangle .

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Contains Method (Int32, Int32)

Determines whether this [Rectangle](#) contains a specified point represented by its x- and y-coordinates.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Contains (  
    int x,  
    int y  
)
```

Parameters

x

The x-coordinate of the specified point.

y

The y-coordinate of the specified point.

Return Value

true if the specified point is contained within this [Rectangle](#); **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Contains Method (Point)

Determines whether this [Rectangle](#) contains a specified [Point](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Contains (  
    Point value  
)
```

Parameters

value

The [Point](#) to evaluate.

Return Value

true if the specified [Point](#) is contained within this [Rectangle](#); **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Contains Method (Point, Boolean)

Determines whether this [Rectangle](#) contains a specified [Point](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref Point value,  
    out bool result  
)
```

Parameters

value

The [Point](#) to evaluate.

result

[[OutAttribute](#)] **true** if the specified [Point](#) is contained within this [Rectangle](#); **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Contains Method (Rectangle)

Determines whether this [Rectangle](#) entirely contains a specified [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Contains (  
    Rectangle value  
)
```

Parameters

value

The [Rectangle](#) to evaluate.

Return Value

true if this [Rectangle](#) entirely contains the specified [Rectangle](#); **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Contains Method (Rectangle, Boolean)

Determines whether this [Rectangle](#) entirely contains a specified [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Contains (  
    ref Rectangle value,  
    out bool result  
)
```

Parameters

value

The [Rectangle](#) to evaluate.

result

[[OutAttribute](#)] On exit, is **true** if this [Rectangle](#) entirely contains the specified [Rectangle](#), or **false** if not.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Rectangle.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Rectangle.Equals (Rectangle)	Determines whether the specified Object is equal to the Rectangle .
Rectangle.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object to make the comparison with.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Equals Method (Rectangle)

Determines whether the specified [Object](#) is equal to the [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Rectangle other  
)
```

Parameters

other

The [Object](#) to compare with the current [Rectangle](#).

Return Value

true if the specified [Object](#) is equal to the current [Rectangle](#); **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Inflate Method

Pushes the edges of the [Rectangle](#) out by the horizontal and vertical values specified.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Inflate (  
    int horizontalAmount,  
    int verticalAmount  
)
```

Parameters

horizontalAmount

Value to push the sides out by.

verticalAmount

Value to push the top and bottom out by.

Remarks

Each corner of the [Rectangle](#) is pushed away from the center of the rectangle by the specified amounts. This results in the width and height of the [Rectangle](#) increasing by twice the values provided.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Intersect Method

Creates a [Rectangle](#) defining the area where one rectangle overlaps another rectangle.

Overload List

Name	Description
Rectangle.Intersect (Rectangle, Rectangle)	Creates a Rectangle defining the area where one rectangle overlaps with another rectangle.
Rectangle.Intersect (Rectangle, Rectangle, Rectangle)	Creates a Rectangle defining the area where one rectangle overlaps with another rectangle.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Intersect Method (Rectangle, Rectangle)

Creates a [Rectangle](#) defining the area where one rectangle overlaps with another rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Rectangle Intersect (  
    Rectangle value1,  
    Rectangle value2  
)
```

Parameters

value1

The first [Rectangle](#) to compare.

value2

The second [Rectangle](#) to compare.

Return Value

The area where the two parameters overlap.

Remarks

If the two rectangles do not overlap, the resulting [Rectangle](#) will be empty.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Intersect Method (Rectangle, Rectangle, Rectangle)

Creates a [Rectangle](#) defining the area where one rectangle overlaps with another rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Intersect (  
    ref Rectangle value1,  
    ref Rectangle value2,  
    out Rectangle result  
)
```

Parameters

value1

The first [Rectangle](#) to compare.

value2

The second [Rectangle](#) to compare.

result

[[OutAttribute](#)] The area where the two first parameters overlap.

Remarks

If the two rectangles do not overlap, the resulting [Rectangle](#) will be empty.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Intersects Method

Determines whether a specified [Rectangle](#) intersects with this [Rectangle](#).

Overload List

Name	Description
Rectangle.Intersects (Rectangle)	Determines whether a specified Rectangle intersects with this Rectangle .
Rectangle.Intersects (Rectangle, Boolean)	Determines whether a specified Rectangle intersects with this Rectangle .

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Intersects Method (Rectangle)

Determines whether a specified [Rectangle](#) intersects with this [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Intersects (  
    Rectangle value  
)
```

Parameters

value

The [Rectangle](#) to evaluate.

Return Value

true if the specified [Rectangle](#) intersects with this one; **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Intersects Method (Rectangle, Boolean)

Determines whether a specified [Rectangle](#) intersects with this [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Intersects (  
    ref Rectangle value,  
    out bool result  
)
```

Parameters

value

The [Rectangle](#) to evaluate

result

[[OutAttribute](#)] **true** if the specified [Rectangle](#) intersects with this one; **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Offset Method

Changes the position of the [Rectangle](#).

Overload List

Name	Description
Rectangle.Offset (Int32, Int32)	Changes the position of the Rectangle .
Rectangle.Offset (Point)	Changes the position of the Rectangle .

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Offset Method (Int32, Int32)

Changes the position of the [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Offset (  
    int offsetX,  
    int offsetY  
)
```

Parameters

offsetX

Change in the x-position.

offsetY

Change in the y-position.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Offset Method (Point)

Changes the position of the [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Offset (  
    Point amount  
)
```

Parameters

amount

The values to adjust the position of the [Rectangle](#) by.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.op_Equality Method

Compares two rectangles for equality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Rectangle a,  
    Rectangle b  
)
```

Parameters

a
Source rectangle.

b
Source rectangle.

Return Value

true if the rectangles are equal; **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.op_Inequality Method

Compares two rectangles for inequality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Rectangle a,  
    Rectangle b  
)
```

Parameters

a

Source rectangle.

b

Source rectangle.

Return Value

true if the rectangles are not equal; **false** otherwise.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.ToString Method

Retrieves a string representation of the current object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Union Method

Creates a new [Rectangle](#) that exactly contains two other rectangles.

Overload List

Name	Description
Rectangle.Union (Rectangle, Rectangle)	Creates a new Rectangle that exactly contains two other rectangles.
Rectangle.Union (Rectangle, Rectangle, Rectangle)	Creates a new Rectangle that exactly contains two other rectangles.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Union Method (Rectangle, Rectangle)

Creates a new [Rectangle](#) that exactly contains two other rectangles.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Rectangle Union (  
    Rectangle value1,  
    Rectangle value2  
)
```

Parameters

value1

The first [Rectangle](#) to contain.

value2

The second [Rectangle](#) to contain.

Return Value

The union of the two [Rectangle](#) parameters.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Union Method (Rectangle, Rectangle, Rectangle)

Creates a new [Rectangle](#) that exactly contains two other rectangles.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Union (  
    ref Rectangle value1,  
    ref Rectangle value2,  
    out Rectangle result  
)
```

Parameters

value1

The first [Rectangle](#) to contain.

value2

The second [Rectangle](#) to contain.

result

[[OutAttribute](#)] The [Rectangle](#) that must be the union of the first two rectangles.

See Also

Reference

[Rectangle Structure](#)









[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle Properties

Public Properties

	Name	Description
	Bottom	Returns the y-coordinate of the bottom of the rectangle.
	Center	Gets the Point that specifies the center of the rectangle.
 S	Empty	Returns a Rectangle with all of its values set to zero.
	IsEmpty	Gets a value that indicates whether the Rectangle is empty.
	Left	Returns the x-coordinate of the left side of the rectangle.
	Location	Gets or sets the upper-left value of the Rectangle .
	Right	Returns the x-coordinate of the right side of the rectangle.
	Top	Returns the y-coordinate of the top of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Rectangle.Bottom Property

Returns the y-coordinate of the bottom of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Bottom { get; }
```

Property Value

The y-coordinate of the bottom of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Center Property

Gets the [Point](#) that specifies the center of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Point Center { get; }
```

Property Value

The center of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Empty Property

Returns a [Rectangle](#) with all of its values set to zero.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Rectangle Empty { get; }
```

Property Value

An empty [Rectangle](#).

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.IsEmpty Property

Gets a value that indicates whether the [Rectangle](#) is empty.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsEmpty { get; }
```

Property Value

true if the rectangle is empty, otherwise **false**.

Remarks

An empty rectangle has all its values set to 0.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Left Property

Returns the x-coordinate of the left side of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Left { get; }
```

Property Value

The x-coordinate of the left side of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Location Property

Gets or sets the upper-left value of the [Rectangle](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Point Location { get; set; }
```

Property Value

The new upper-left corner of the [Rectangle](#).

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Right Property

Returns the x-coordinate of the right side of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Right { get; }
```

Property Value

The x-coordinate of the right side of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rectangle.Top Property

Returns the y-coordinate of the top of the rectangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Top { get; }
```

Property Value

The y-coordinate of the top of the rectangle.

See Also

Reference

[Rectangle Structure](#)

[Rectangle Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

TargetPlatform Enumeration

Note

This enumeration is available only when developing for Windows.

Defines the target platform to be used when compiling content.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum TargetPlatform
```

Members

Member name	Description
Unknown	Represents an unknown platform.
Windows	The Windows platform.
Xbox360	The Xbox 360 game console.
Zune	A Zune device.

See Also

Reference

[BuildContent.TargetPlatform Property](#)

[BuildXact.TargetPlatform Property](#)

[ContentProcessorContext.TargetPlatform Property](#)

[ContentWriter.TargetPlatform Property](#)

[ContentTypeWriter.GetRuntimeType Method](#)

[ContentTypeWriter.GetRuntimeReader Method](#)

[Effect.CompileEffectFromFile Method](#)

[ShaderCompiler.AssembleFromFile Method](#)

[ShaderCompiler.CompileFromFile Method](#)

[ShaderCompiler.CompileFromSource Method](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Write a Custom Importer and Processor](#)

PlatformsWindows XP SP2, Windows Vista

Vector2 Structure

Defines a vector with two components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.Vector2Converter)")]
[SerializableAttribute]
public struct Vector2 : IEquatable<Vector2>
```

See Also

Reference

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Vector2 Members

The following tables list the members exposed by the Vector2 type.





Public Constructors

	Name	Description
	Vector2	Overloaded. Initializes a new instance of Vector2 .
















Public Fields




















	Name	Description
	X	Gets or sets the x-component of the vector.
	Y	Gets or sets the y-component of the vector.

Public Properties



	Name	Description
	One	Returns a Vector2 with both of its components set to one.
	UnitX	Returns the unit vector for the x-axis.
	UnitY	Returns the unit vector for the y-axis.
	Zero	Returns a Vector2 with all of its components set to zero.

Public Methods

	Name	Description
	Add	Overloaded. Adds two vectors.
	Barycentric	Overloaded. Returns a Vector2 containing the 2D Cartesian coordinates of a point specified in 2D barycentric (areal) coordinates.
	CatmullRom	Overloaded. Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Overloaded. Restricts a value to be within a specified range.
	Distance	Overloaded. Calculates the distance between two vectors.
	DistanceSquared	Overloaded. Calculates the distance between two vectors squared.
	Divide	Overloaded. Divides a vector by a scalar or another vector.
	Dot	Overloaded. Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of the vector object.
	GetType	(Inherited from Object .)
	Hermite	Overloaded. Performs a Hermite spline interpolation.
	Length	Calculates the length of the vector.
	LengthSquared	Calculates the length of the vector squared.
	Lerp	Overloaded. Performs a linear interpolation between two vectors.

 Max	Overloaded. Returns a vector that contains the highest value from each matching pair of components.
 Min	Overloaded. Returns a vector that contains the lowest value from each matching pair of components.
 Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
 Negate	Overloaded. Returns a vector pointing in the opposite direction.
 Normalize	Overloaded. Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.
 op_Addition	Adds two vectors.
 op_Division	Overloaded. Divide a vector by a scalar or another vector.
 op_Equality	Tests vectors for equality.
 op_Inequality	Tests vectors for inequality.
 op_Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
 op_Subtraction	Subtracts a vector from a vector.
 op_UnaryNegation	Returns a vector pointing in the opposite direction.
 ReferenceEquals	(Inherited from Object .)
 Reflect	Overloaded. Determines the reflect vector of the given vector and normal.
 SmoothStep	Overloaded. Interpolates between two values using a cubic equation.
 Subtract	Overloaded. Subtracts a vector from a vector.
 ToString	Retrieves a string representation of the current object.
 Transform	Overloaded. Transforms one or more Vector2s by a Matrix or Quaternion .
 TransformNormal	Overloaded. Transforms a vector normal or array of vector normals by a matrix.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[Vector2 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2 Fields

Public Fields

	Name	Description
	X	Gets or sets the x-component of the vector.
	Y	Gets or sets the y-component of the vector.

See Also

Reference

[Vector2 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.X Field

Gets or sets the x-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float X
```

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Y Field

Gets or sets the y-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Y
```

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2 Constructor

Initializes a new instance of [Vector2](#).

Overload List

Name	Description
Vector2 (Single)	Creates a new instance of Vector2 .
Vector2 (Single, Single)	Initializes a new instance of Vector2 .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2 Constructor (Single)

Creates a new instance of [Vector2](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 (  
    float value  
)
```

Parameters

value

Value to initialize both components to.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2 Constructor (Single, Single)

Initializes a new instance of [Vector2](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 (  
    float x,  
    float y  
)
```

Parameters

x

Initial value for the x-component of the vector.

y

Initial value for the y-component of the vector.

See Also

Reference

[Vector2 Structure](#)























[Vector2 Members](#)













[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Vector2 Methods

Public Methods

	Name	Description
	Add	Overloaded. Adds two vectors.
	Barycentric	Overloaded. Returns a Vector2 containing the 2D Cartesian coordinates of a point specified in 2D barycentric (areal) coordinates.
	CatmullRom	Overloaded. Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Overloaded. Restricts a value to be within a specified range.
	Distance	Overloaded. Calculates the distance between two vectors.
	DistanceSquared	Overloaded. Calculates the distance between two vectors squared.
	Divide	Overloaded. Divides a vector by a scalar or another vector.
	Dot	Overloaded. Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of the vector object.
	GetType	(Inherited from Object .)
	Hermite	Overloaded. Performs a Hermite spline interpolation.
	Length	Calculates the length of the vector.
	LengthSquared	Calculates the length of the vector squared.
	Lerp	Overloaded. Performs a linear interpolation between two vectors.
	Max	Overloaded. Returns a vector that contains the highest value from each matching pair of components.
	Min	Overloaded. Returns a vector that contains the lowest value from each matching pair of components.
	Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
	Negate	Overloaded. Returns a vector pointing in the opposite direction.
	Normalize	Overloaded. Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.
	op_Addition	Adds two vectors.
	op_Division	Overloaded. Divide a vector by a scalar or another vector.

 op_Equality	Tests vectors for equality.
 op_Inequality	Tests vectors for inequality.
 op_Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
 op_Subtraction	Subtracts a vector from a vector.
 op_UnaryNegation	Returns a vector pointing in the opposite direction.
 ReferenceEquals	(Inherited from Object .)
 Reflect	Overloaded. Determines the reflect vector of the given vector and normal.
 SmoothStep	Overloaded. Interpolates between two values using a cubic equation.
 Subtract	Overloaded. Subtracts a vector from a vector.
 ToString	Retrieves a string representation of the current object.
 Transform	Overloaded. Transforms one or more Vector2 s by a Matrix or Quaternion .
 TransformNormal	Overloaded. Transforms a vector normal or array of vector normals by a matrix.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Vector2 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Add Method

Adds two vectors.

Overload List

Name	Description
Vector2.Add (Vector2, Vector2)	Adds two vectors.
Vector2.Add (Vector2, Vector2, Vector2)	Adds two vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Add Method (Vector2, Vector2)

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Add (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Sum of the source vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Add Method (Vector2, Vector2, Vector2)

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Add (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] Sum of the source vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Barycentric Method

Returns a [Vector2](#) containing the 2D Cartesian coordinates of a point specified in 2D barycentric (areal) coordinates.

Overload List

Name	Description
Vector2.Barycentric (Vector2, Vector2, Vector2, Single, Single)	Returns a Vector2 containing the 2D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 2D triangle.
Vector2.Barycentric (Vector2, Vector2, Vector2, Single, Single, Vector2)	Returns a Vector2 containing the 2D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 2D triangle.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Barycentric Method (Vector2, Vector2, Vector2, Single, Single)

Returns a [Vector2](#) containing the 2D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 2D triangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Barycentric (
    Vector2 value1,
    Vector2 value2,
    Vector2 value3,
    float amount1,
    float amount2
)
```

Parameters

value1

A [Vector2](#) containing the 2D Cartesian coordinates of vertex 1 of the triangle.

value2

A [Vector2](#) containing the 2D Cartesian coordinates of vertex 2 of the triangle.

value3

A [Vector2](#) containing the 2D Cartesian coordinates of vertex 3 of the triangle.

amount1

Barycentric coordinate **b2**, which expresses the weighting factor toward vertex 2 (specified in *value2*).

amount2

Barycentric coordinate **b3**, which expresses the weighting factor toward vertex 3 (specified in *value3*).

Return Value

A new [Vector2](#) containing the 2D Cartesian coordinates of the specified point.

Remarks

About Barycentric Coordinates

Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by three weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas.

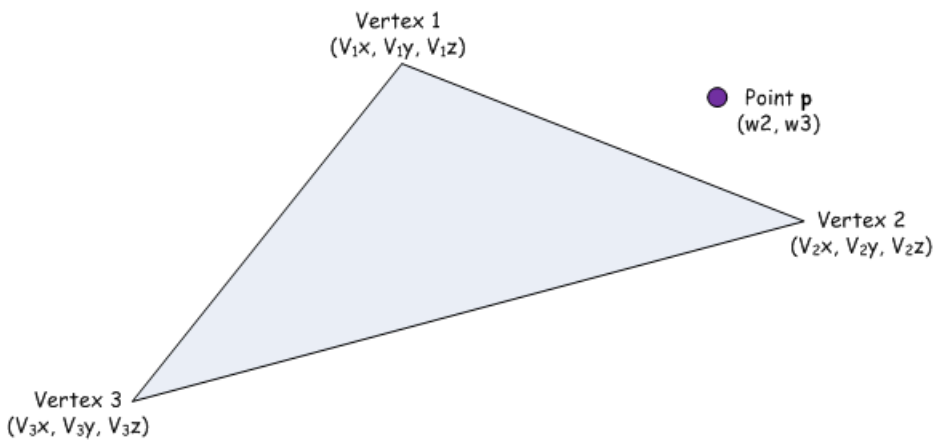
$$P_x = (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x});$$

$$P_y = (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y});$$

Such triple weighting factors **b1**, **b2**, and **b3** are called *barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that **(k * b1)**, **(k * b2)**, and **(k * b3)** are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of barycentric coordinates is normalized so that **b1 + b2 + b3 = 1**, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** equals **(1 - b2 - b3)**.



What Vector2 Barycentric Does

The **Vector2 Barycentric** method takes three vectors specifying the Cartesian coordinates of the triangle vertices, **V1**, **V2**, and **V3**, and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument, and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

Barycentric then calculates the Cartesian coordinate of **P** as follows:

$$P_x = ((1 - b_2 - b_3) * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}) ;$$

$$P_y = ((1 - b_2 - b_3) * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}) ;$$

Thus, to calculate the 2D Cartesian coordinates of **P**, you would pass the coordinates of the triangle vertices to **Barycentric** together with the appropriate normalized barycentric coordinates of **P**.

The following relationships may be useful.

- If (*amount1* <= 0) and (*amount2* >= 0) and ($1 - \text{amount1} - \text{amount2} >= 0$), then the point is inside the triangle defined by *value1*, *value2*, and *value3*.
- If (*amount1* == 0) and (*amount2* >= 0) and ($1 - \text{amount1} - \text{amount2} >= 0$), then the point is on the line defined by *value1* and *value3*.
- If (*amount1* >= 0) and (*amount2* == 0) and ($1 - \text{amount1} - \text{amount2} >= 0$), then the point is on the line defined by *value1* and *value2*.
- If (*amount1* >= 0) and (*amount2* >= 0) and ($1 - \text{amount1} - \text{amount2} == 0$), then the point is on the line defined by *value2* and *value3*.

Barycentric coordinates are a form of general coordinates. In this context, using barycentric coordinates represents a change in coordinate systems. What holds true for Cartesian coordinates holds true for barycentric coordinates.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Barycentric Method (Vector2, Vector2, Vector2, Single, Single, Vector2)

Returns a [Vector2](#) containing the 2D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 2D triangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Barycentric (
    ref Vector2 value1,
    ref Vector2 value2,
    ref Vector2 value3,
    float amount1,
    float amount2,
    out Vector2 result
)
```

Parameters

value1

A [Vector2](#) containing the 2D Cartesian coordinates of vertex 1 of the triangle.

value2

A [Vector2](#) containing the 2D Cartesian coordinates of vertex 2 of the triangle.

value3

A [Vector2](#) containing the 2D Cartesian coordinates of vertex 3 of the triangle.

amount1

Barycentric coordinate **b2**, which expresses the weighting factor toward vertex 2 (specified in *value2*).

amount2

Barycentric coordinate **b3**, which expresses the weighting factor toward vertex 3 (specified in *value3*).

result

[[OutAttribute](#)] The 2D Cartesian coordinates of the specified point are placed in this [Vector2](#) on exit.

Remarks

About Barycentric Coordinates

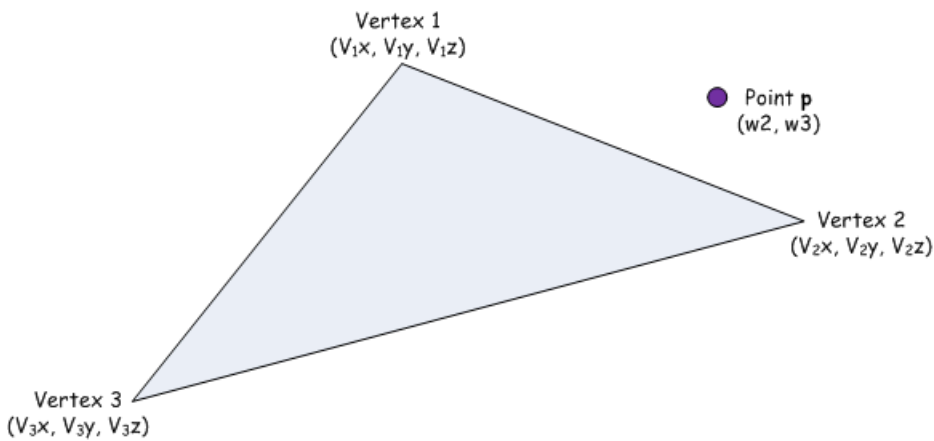
Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by three weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas.

$$\begin{aligned} P_x &= (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \end{aligned}$$

Such triple weighting factors **b1**, **b2**, and **b3** are called *barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that **(k * b1)**, **(k * b2)**, and **(k * b3)** are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of barycentric coordinates is normalized so that **b1 + b2 + b3 = 1**, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** equals **(1 - b2 - b3)**.



What Vector2 Barycentric Does

The **Vector2 Barycentric** method takes three vectors specifying the Cartesian coordinates of the triangle vertices, **V1**, **V2**, and **V3**, and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument, and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

Barycentric then calculates the Cartesian coordinate of **P** as follows:

$$P_x = ((1 - b_2 - b_3) * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}) ;$$

$$P_y = ((1 - b_2 - b_3) * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}) ;$$

Thus, to calculate the 2D Cartesian coordinates of **P**, you would pass the coordinates of the triangle vertices to **Barycentric** together with the appropriate normalized barycentric coordinates of **P**.

The following relationships may be useful.

- If (*amount1* <= 0) and (*amount2* >= 0) and ($1 - \text{amount1} - \text{amount2} >= 0$), then the point is inside the triangle defined by *value1*, *value2*, and *value3*.
- If (*amount1* == 0) and (*amount2* >= 0) and ($1 - \text{amount1} - \text{amount2} >= 0$), then the point is on the line defined by *value1* and *value3*.
- If (*amount1* >= 0) and (*amount2* == 0) and ($1 - \text{amount1} - \text{amount2} >= 0$), then the point is on the line defined by *value1* and *value2*.
- If (*amount1* >= 0) and (*amount2* >= 0) and ($1 - \text{amount1} - \text{amount2} == 0$), then the point is on the line defined by *value2* and *value3*.

Barycentric coordinates are a form of general coordinates. In this context, using barycentric coordinates represents a change in coordinate systems. What holds true for Cartesian coordinates holds true for barycentric coordinates.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.CatmullRom Method

Performs a Catmull-Rom interpolation using the specified positions.

Overload List

Name	Description
Vector2.CatmullRom (Vector2, Vector2, Vector2, Vector2, Single)	Performs a Catmull-Rom interpolation using the specified positions.
Vector2.CatmullRom (Vector2, Vector2, Vector2, Vector2, Single, Vector2)	Performs a Catmull-Rom interpolation using the specified positions.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.CatmullRom Method (Vector2, Vector2, Vector2, Vector2, Single)

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 CatmullRom (  
    Vector2 value1,  
    Vector2 value2,  
    Vector2 value3,  
    Vector2 value4,  
    float amount  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

Return Value

A vector that is the result of the Catmull-Rom interpolation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.CatmullRom Method (Vector2, Vector2, Vector2, Vector2, Single, Vector2)

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CatmullRom (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    ref Vector2 value3,  
    ref Vector2 value4,  
    float amount,  
    out Vector2 result  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

result

[[OutAttribute](#)] A vector that is the result of the Catmull-Rom interpolation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Clamp Method

Restricts a value to be within a specified range.

Overload List

Name	Description
Vector2.Clamp (Vector2, Vector2, Vector2)	Restricts a value to be within a specified range.
Vector2.Clamp (Vector2, Vector2, Vector2, Vector2)	Restricts a value to be within a specified range.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Clamp Method (Vector2, Vector2, Vector2)

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Clamp (  
    Vector2 value1,  
    Vector2 min,  
    Vector2 max  
)
```

Parameters

value1

The value to clamp.

min

The minimum value.

max

The maximum value.

Return Value

The clamped value.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Clamp Method (Vector2, Vector2, Vector2, Vector2)

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Clamp (  
    ref Vector2 value1,  
    ref Vector2 min,  
    ref Vector2 max,  
    out Vector2 result  
)
```

Parameters

value1

The value to clamp.

min

The minimum value.

max

The maximum value.

result

[[OutAttribute](#)] The clamped value.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Distance Method

Calculates the distance between two vectors.

Overload List

Name	Description
Vector2.Distance (Vector2, Vector2)	Calculates the distance between two vectors.
Vector2.Distance (Vector2, Vector2, Single)	Calculates the distance between two vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Distance Method (Vector2, Vector2)

Calculates the distance between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Distance (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Distance between the two vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Distance Method (Vector2, Vector2, Single)

Calculates the distance between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Distance (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out float result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The distance between the vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.DistanceSquared Method

Calculates the distance between two vectors squared.

Overload List

Name	Description
Vector2.DistanceSquared (Vector2, Vector2)	Calculates the distance between two vectors squared.
Vector2.DistanceSquared (Vector2, Vector2, Single)	Calculates the distance between two vectors squared.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.DistanceSquared Method (Vector2, Vector2)

Calculates the distance between two vectors squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float DistanceSquared (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The distance between the source vectors squared.

Remarks

Distance is calculated with the formula:

$$\sqrt{(a - b)^2}$$

Distance squared is the value before taking the square root. Distance squared can often be used in place of distance if relative comparisons are being made. For example, consider three points A, B, and C. To determine whether B or C is further from A, compare the distance between A and B to the distance between A and C. Calculating the two distances involves two square roots, which are computationally expensive. However, using distance squared provides the same information and avoids calculating two square roots.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.DistanceSquared Method (Vector2, Vector2, Single)

Calculates the distance between two vectors squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void DistanceSquared (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out float result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The distance between the vectors squared.

Remarks

Distance is calculated with the formula:

$$\sqrt{(a - b)^2}$$

Distance squared is the value before taking the square root. Distance squared can often be used in place of distance if relative comparisons are being made. For example, consider three points A, B, and C. To determine whether B or C is further from A, compare the distance between A and B to the distance between A and C. Calculating the two distances involves two square roots, which are computationally expensive. However, using distance squared provides the same information and avoids calculating two square roots.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Divide Method

Divides a vector by a scalar or another vector.

Overload List

Name	Description
Vector2.Divide (Vector2, Single)	Divides a vector by a scalar value.
Vector2.Divide (Vector2, Single, Vector2)	Divides a vector by a scalar value.
Vector2.Divide (Vector2, Vector2)	Divides the components of a vector by the components of another vector.
Vector2.Divide (Vector2, Vector2, Vector2)	Divides the components of a vector by the components of another vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Divide Method (Vector2, Single)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Divide (  
    Vector2 value1,  
    float divider  
)
```

Parameters

value1

Source vector.

divider

The divisor.

Return Value

The result of the division.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Divide Method (Vector2, Single, Vector2)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Vector2 value1,  
    float divider,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

divider

The divisor.

result

[[OutAttribute](#)] The result of the division.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Divide Method (Vector2, Vector2)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Divide (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Divisor vector.

Return Value

The result of dividing the vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Divide Method (Vector2, Vector2, Vector2)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

The divisor.

result

[[OutAttribute](#)] The result of the division.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Dot Method

Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Overload List

Name	Description
Vector2.Dot (Vector2, Vector2)	Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.
Vector2.Dot (Vector2, Vector2, Single)	Calculates the dot product of two vectors and writes the result to a user-specified variable. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Remarks

The dot product is also known as the inner product.

For any two vectors, the dot product is defined as:

$$(\text{vector1.X} * \text{vector2.X}) + (\text{vector1.Y} * \text{vector2.Y})$$

The result of this calculation, plus or minus some margin to account for floating point error, is equal to:

$$\text{Length}(\text{vector1}) * \text{Length}(\text{vector2}) * \text{System.Math.Cos}(\text{theta})$$

Here, *theta* is the angle between the two vectors.

If *vector1* and *vector2* are unit vectors, the length of each vector will be equal to 1. So, when *vector1* and *vector2* are unit vectors, the dot product is simply equal to the cosine of the angle between the two vectors. The following calculation:

$$\text{Vector2.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}())$$

Is equivalent to:

$$\text{System.Math.Cos}(\text{theta})$$

Therefore, if *vector1* and *vector2* are unit vectors (i.e., we've called [Normalize](#) on each vector to make it a unit vector), without knowing the value of *theta* or using a potentially processor-intensive trigonometric function, the dot product can tell us the following things:

- If $\text{Vector2.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) > 0$, the angle between the two vectors is less than 90 degrees.
- If $\text{Vector2.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) < 0$, the angle between the two vectors is more than 90 degrees.
- If $\text{Vector2.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == 0$, the angle between the two vectors is 90 degrees; that is, the vectors are orthogonal.
- If $\text{Vector2.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == 1$, the angle between the two vectors is 0 degrees; that is, the vectors point in the same direction and are parallel.
- If $\text{Vector2.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == -1$, the angle between the two vectors is 180 degrees; that is, the vectors point in opposite directions and are parallel.

⚠Caution

Because of floating point error, two orthogonal vectors may not return a dot product that is exactly zero. It might be zero plus some amount of floating point error. In your code, you will want to determine what amount of error is acceptable in your calculation, and take that into account when you do your comparisons.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Dot Method (Vector2, Vector2)

Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Dot (
    Vector2 value1,
    Vector2 value2
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The dot product of the two vectors.

Remarks

The dot product is also known as the inner product.

For any two vectors, the dot product is defined as:

$$(\text{vector1.X} * \text{vector2.X}) + (\text{vector1.Y} * \text{vector2.Y})$$

The result of this calculation, plus or minus some margin to account for floating point error, is equal to:

$$\text{Length}(\text{vector1}) * \text{Length}(\text{vector2}) * \text{System.Math.Cos}(\text{theta})$$

Here, *theta* is the angle between the two vectors.

If *vector1* and *vector2* are unit vectors, the length of each vector will be equal to 1. So, when *vector1* and *vector2* are unit vectors, the dot product is simply equal to the cosine of the angle between the two vectors. The following calculation:

```
Vector2.Dot( vector1.Normalize(), vector2.Normalize() )
```

Is equivalent to:

```
System.Math.Cos( theta )
```

Therefore, if *vector1* and *vector2* are unit vectors (i.e., we've called [Normalize](#) on each vector to make it a unit vector), without knowing the value of *theta* or using a potentially processor-intensive trigonometric function, the dot product can tell us the following things:

- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) > 0`, the angle between the two vectors is less than 90 degrees.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) < 0`, the angle between the two vectors is more than 90 degrees.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) == 0`, the angle between the two vectors is 90 degrees; that is, the vectors are orthogonal.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) == 1`, the angle between the two vectors is 0 degrees; that is, the vectors point in the same direction and are parallel.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) == -1`, the angle between the two vectors is 180 degrees; that is, the vectors point in opposite directions and are parallel.

⚠ Caution

Because of floating point error, two orthogonal vectors may not return a dot product that is exactly zero. It might be zero plus some amount of floating point error. In your code, you will want to determine what amount of error is acceptable in your calculation, and take that into account when you do your comparisons.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Dot Method (Vector2, Vector2, Single)

Calculates the dot product of two vectors and writes the result to a user-specified variable. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Dot (
    ref Vector2 value1,
    ref Vector2 value2,
    out float result
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The dot product of the two vectors.

Remarks

The dot product is also known as the inner product.

For any two vectors, the dot product is defined as:

$$(\text{vector1.X} * \text{vector2.X}) + (\text{vector1.Y} * \text{vector2.Y})$$

The result of this calculation, plus or minus some margin to account for floating point error, is equal to:

$$\text{Length}(\text{vector1}) * \text{Length}(\text{vector2}) * \text{System.Math.Cos}(\text{theta})$$

Here, *theta* is the angle between the two vectors.

If *vector1* and *vector2* are unit vectors, the length of each vector will be equal to 1. So, when *vector1* and *vector2* are unit vectors, the dot product is simply equal to the cosine of the angle between the two vectors. The following calculation:

```
Vector2.Dot( vector1.Normalize(), vector2.Normalize() )
```

Is equivalent to:

```
System.Math.Cos( theta )
```

Therefore, if *vector1* and *vector2* are unit vectors (i.e., we've called [Normalize](#) on each vector to make it a unit vector), without knowing the value of *theta* or using a potentially processor-intensive trigonometric function, the dot product can tell us the following things:

- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) > 0`, the angle between the two vectors is less than 90 degrees.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) < 0`, the angle between the two vectors is more than 90 degrees.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) == 0`, the angle between the two vectors is 90 degrees; that is, the vectors are orthogonal.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) == 1`, the angle between the two vectors is 0 degrees; that is, the vectors point in the same direction and are parallel.
- If `Vector2.Dot(vector1.Normalize(), vector2.Normalize()) == -1`, the angle between the two vectors is 180 degrees; that is, the vectors point in opposite directions and are parallel.

⚠ Caution

Because of floating point error, two orthogonal vectors may not return a dot product that is exactly zero. It might be zero plus some amount of floating point error. In your code, you will want to determine what amount of error is acceptable in your calculation, and take that into account when you do your comparisons.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Vector2.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Vector2.Equals (Vector2)	Determines whether the specified Object is equal to the Vector2 .
Vector2.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object to make the comparison with.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Equals Method (Vector2)

Determines whether the specified [Object](#) is equal to the [Vector2](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Vector2 other  
)
```

Parameters

other

The [Object](#) to compare with the current [Vector2](#).

Return Value

true if the specified [Object](#) is equal to the current [Vector2](#); **false** otherwise.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.GetHashCode Method

Gets the hash code of the vector object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code of the vector object.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Hermite Method

Performs a Hermite spline interpolation.

Overload List

Name	Description
Vector2.Hermite (Vector2, Vector2, Vector2, Vector2, Single)	Performs a Hermite spline interpolation.
Vector2.Hermite (Vector2, Vector2, Vector2, Vector2, Single, Vector2)	Performs a Hermite spline interpolation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Hermite Method (Vector2, Vector2, Vector2, Vector2, Single)

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Hermite (  
    Vector2 value1,  
    Vector2 tangent1,  
    Vector2 value2,  
    Vector2 tangent2,  
    float amount  
)
```

Parameters

value1

Source position vector.

tangent1

Source tangent vector.

value2

Source position vector.

tangent2

Source tangent vector.

amount

Weighting factor.

Return Value

The result of the Hermite spline interpolation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Hermite Method (Vector2, Vector2, Vector2, Vector2, Single, Vector2)

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Hermite (  
    ref Vector2 value1,  
    ref Vector2 tangent1,  
    ref Vector2 value2,  
    ref Vector2 tangent2,  
    float amount,  
    out Vector2 result  
)
```

Parameters

value1

Source position vector.

tangent1

Source tangent vector.

value2

Source position vector.

tangent2

Source tangent vector.

amount

Weighting factor.

result

[[OutAttribute](#)] The result of the Hermite spline interpolation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Length Method

Calculates the length of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Length ()
```

Return Value

Length of the vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.LengthSquared Method

Calculates the length of the vector squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float LengthSquared ()
```

Return Value

The length of the vector squared.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Lerp Method

Performs a linear interpolation between two vectors.

Overload List

Name	Description
Vector2.Lerp (Vector2, Vector2, Single)	Performs a linear interpolation between two vectors.
Vector2.Lerp (Vector2, Vector2, Single, Vector2)	Performs a linear interpolation between two vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Lerp Method (Vector2, Vector2, Single)

Performs a linear interpolation between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Lerp (  
    Vector2 value1,  
    Vector2 value2,  
    float amount  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Value between 0 and 1 indicating the weight of *value2*.

Return Value

The linear interpolation of the two vectors.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned; a value of 1 will cause *value2* to be returned.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Lerp Method (Vector2, Vector2, Single, Vector2)

Performs a linear interpolation between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Lerp (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    float amount,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Value between 0 and 1 indicating the weight of *value2*.

result

[[OutAttribute](#)] The result of the interpolation.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned; a value of 1 will cause *value2* to be returned.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Max Method

Returns a vector that contains the highest value from each matching pair of components.

Overload List

Name	Description
Vector2.Max (Vector2, Vector2)	Returns a vector that contains the highest value from each matching pair of components.
Vector2.Max (Vector2, Vector2, Vector2)	Returns a vector that contains the highest value from each matching pair of components.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Max Method (Vector2, Vector2)

Returns a vector that contains the highest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Max (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The maximized vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Max Method (Vector2, Vector2, Vector2)

Returns a vector that contains the highest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Max (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The maximized vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Min Method

Returns a vector that contains the lowest value from each matching pair of components.

Overload List

Name	Description
Vector2.Min (Vector2, Vector2)	Returns a vector that contains the lowest value from each matching pair of component s.
Vector2.Min (Vector2, Vector2, Vector2)	Returns a vector that contains the lowest value from each matching pair of component s.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Min Method (Vector2, Vector2)

Returns a vector that contains the lowest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Min (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The minimized vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Min Method (Vector2, Vector2, Vector2)

Returns a vector that contains the lowest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Min (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The minimized vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Multiply Method

Multiplies a vector by a scalar or another vector.

Overload List

Name	Description
Vector2.Multiply (Vector2, Single)	Multiplies a vector by a scalar value.
Vector2.Multiply (Vector2, Single, Vector2)	Multiplies a vector by a scalar value.
Vector2.Multiply (Vector2, Vector2)	Multiplies the components of two vectors by each other.
Vector2.Multiply (Vector2, Vector2, Vector2)	Multiplies the components of two vectors by each other.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Multiply Method (Vector2, Single)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Multiply (  
    Vector2 value1,  
    float scaleFactor  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Multiply Method (Vector2, Single, Vector2)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Vector2 value1,  
    float scaleFactor,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Multiply Method (Vector2, Vector2)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Multiply (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the multiplication.

Remarks

Multiplication performed by this method is not vector multiplication (dot product and cross product) but multiplication of the corresponding components of each vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Multiply Method (Vector2, Vector2, Vector2)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Negate Method

Returns a vector pointing in the opposite direction.

Overload List

Name	Description
Vector2.Negate (Vector2)	Returns a vector pointing in the opposite direction.
Vector2.Negate (Vector2, Vector2)	Returns a vector pointing in the opposite direction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Negate Method (Vector2)

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Negate (  
    Vector2 value  
)
```

Parameters

value

Source vector.

Return Value

Vector pointing in the opposite direction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Negate Method (Vector2, Vector2)

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Negate (  
    ref Vector2 value,  
    out Vector2 result  
)
```

Parameters

value

Source vector.

result

[[OutAttribute](#)] Vector pointing in the opposite direction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Normalize Method

Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.

Overload List

Name	Description
Vector2.Normalize ()	Turns the current vector into a unit vector. The result is a vector one unit in length pointing in the same direction as the original vector.
Vector2.Normalize (Vector2)	Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.
Vector2.Normalize (Vector2, Vector2)	Creates a unit vector from the specified vector, writing the result to a user-specified variable. The result is a vector one unit in length pointing in the same direction as the original vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Normalize Method ()

Turns the current vector into a unit vector. The result is a vector one unit in length pointing in the same direction as the original vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Normalize ()
```

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Normalize Method (Vector2)

Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Normalize (  
    Vector2 value  
)
```

Parameters

value

Source [Vector2](#).

Return Value

The created unit vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Normalize Method (Vector2, Vector2)

Creates a unit vector from the specified vector, writing the result to a user-specified variable. The result is a vector one unit in length pointing in the same direction as the original vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Normalize (  
    ref Vector2 value,  
    out Vector2 result  
)
```

Parameters

value

Source vector.

result

[[OutAttribute](#)] Normalized vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Addition Method

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Addition (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Sum of the source vectors.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Division Method

Divide a vector by a scalar or another vector.

Overload List

Name	Description
Vector2.op_Division (Vector2, Single)	Divides a vector by a scalar value.
Vector2.op_Division (Vector2, Vector2)	Divides the components of a vector by the components of another vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.op_Division Method (Vector2, Single)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Division (  
    Vector2 value1,  
    float divider  
)
```

Parameters

value1

Source vector.

divider

The divisor.

Return Value

The source vector divided by *b*.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Division Method (Vector2, Vector2)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Division (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Divisor vector.

Return Value

The result of dividing the vectors.

Remarks

Division of a vector by another vector is not mathematically defined. This method simply divides each component of *a* by the matching component of *b*.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Equality Method

Tests vectors for equality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

true if the vectors are equal; **false** otherwise.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Inequality Method

Tests vectors for inequality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Vector to compare.

value2

Vector to compare.

Return Value

Returns **true** if the vectors are not equal, **false** otherwise.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Multiply Method

Multiplies a vector by a scalar or another vector.

Overload List

Name	Description
Vector2.op_Multiply (Single, Vector2)	Multiplies a vector by a scalar value.
Vector2.op_Multiply (Vector2, Single)	Multiplies a vector by a scalar value.
Vector2.op_Multiply (Vector2, Vector2)	Multiplies the components of two vectors by each other.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.op_Multiply Method (Single, Vector2)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Multiply (  
    float scaleFactor,  
    Vector2 value  
)
```

Parameters

scaleFactor

Scalar value.

value

Source vector.

Return Value

Result of the multiplication.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Multiply Method (Vector2, Single)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Multiply (  
    Vector2 value,  
    float scaleFactor  
)
```

Parameters

value

Source vector.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Multiply Method (Vector2, Vector2)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Multiply (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the multiplication.

Remarks

Multiplication performed by this method is not vector multiplication (dot product and cross product) but multiplication of the corresponding components of each vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_Subtraction Method

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_Subtraction (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

source vector.

Return Value

Result of the subtraction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.op_UnaryNegation Method

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 op_UnaryNegation (  
    Vector2 value  
)
```

Parameters

value

Source vector.

Return Value

Vector pointing in the opposite direction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Reflect Method

Determines the reflect vector of the given vector and normal.

Overload List

Name	Description
Vector2.Reflect (Vector2, Vector2)	Determines the reflect vector of the given vector and normal.
Vector2.Reflect (Vector2, Vector2, Vector2)	Determines the reflect vector of the given vector and normal.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Reflect Method (Vector2, Vector2)

Determines the reflect vector of the given vector and normal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Reflect (  
    Vector2 vector,  
    Vector2 normal  
)
```

Parameters

vector

Source vector.

normal

Normal of *vector*.

Return Value

Resulting reflect vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Reflect Method (Vector2, Vector2, Vector2)

Determines the reflect vector of the given vector and normal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Reflect (  
    ref Vector2 vector,  
    ref Vector2 normal,  
    out Vector2 result  
)
```

Parameters

vector

Source vector.

normal

Normal of *vector*.

result

[[OutAttribute](#)] The created reflect vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.SmoothStep Method

Interpolates between two values using a cubic equation.

Overload List

Name	Description
Vector2.SmoothStep (Vector2, Vector2, Single)	Interpolates between two values using a cubic equation.
Vector2.SmoothStep (Vector2, Vector2, Single, Vector2)	Interpolates between two values using a cubic equation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.SmoothStep Method (Vector2, Vector2, Single)

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 SmoothStep (  
    Vector2 value1,  
    Vector2 value2,  
    float amount  
)
```

Parameters

value1

Source value.

value2

Source value.

amount

Weighting value.

Return Value

Interpolated value.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.SmoothStep Method (Vector2, Vector2, Single, Vector2)

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void SmoothStep (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    float amount,  
    out Vector2 result  
)
```

Parameters

value1

Source value.

value2

Source value.

amount

Weighting value.

result

[[OutAttribute](#)] The interpolated value.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Subtract Method

Subtracts a vector from a vector.

Overload List

Name	Description
Vector2.Subtract (Vector2, Vector2)	Subtracts a vector from a vector.
Vector2.Subtract (Vector2, Vector2, Vector2)	Subtracts a vector from a vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Subtract Method (Vector2, Vector2)

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Subtract (  
    Vector2 value1,  
    Vector2 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the subtraction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Subtract Method (Vector2, Vector2, Vector2)

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Subtract (  
    ref Vector2 value1,  
    ref Vector2 value2,  
    out Vector2 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The result of the subtraction.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.ToString Method

Retrieves a string representation of the current object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method

Transforms one or more [Vector2](#)s by a [Matrix](#) or [Quaternion](#).

Overload List

Name	Description
Vector2.Transform (Vector2, Matrix)	Transforms the vector (x, y, 0, 1) by the specified matrix.
Vector2.Transform (Vector2, Matrix, Vector2)	Transforms a Vector2 by the given Matrix .
Vector2.Transform (Vector2, Quaternion)	Transforms a single Vector2 , or the vector normal (x, y, 0, 0), by a specified Quaternion rotation.
Vector2.Transform (Vector2, Quaternion, Vector2)	Transforms a Vector2 , or the vector normal (x, y, 0, 0), by a specified Quaternion rotation.
Vector2.Transform (Vector2[], Int32, Matrix, Vector2[], Int32, Int32)	Transforms a specified range in an array of Vector2 s by a specified Matrix and places the results in a specified range in a destination array.
Vector2.Transform (Vector2[], Int32, Quaternion, Vector2[], Int32, Int32)	Transforms a specified range in an array of Vector2 s by a specified Quaternion and places the results in a specified range in a destination array.
Vector2.Transform (Vector2[], Matrix, Vector2[])	Transforms an array of Vector2 s by a specified Matrix .
Vector2.Transform (Vector2[], Quaternion, Vector2[])	Transforms an array of Vector2 s by a specified Quaternion .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.Transform Method (Vector2, Matrix)

Transforms the vector (x, y, 0, 1) by the specified matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Transform (  
    Vector2 position,  
    Matrix matrix  
)
```

Parameters

position

The source vector.

matrix

The transformation matrix.

Return Value

The transformed vector.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2, Matrix, Vector2)

Transforms a [Vector2](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector2 position,  
    ref Matrix matrix,  
    out Vector2 result  
)
```

Parameters

position

The source [Vector2](#).

matrix

The transformation [Matrix](#).

result

[[OutAttribute](#)] The [Vector2](#) resulting from the transformation.

Exceptions

Exception type	Condition
ArgumentException	The target array size must be equal or larger than the source array size.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2, Quaternion)

Transforms a single [Vector2](#), or the vector normal (x, y, 0, 0), by a specified [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Transform (  
    Vector2 value,  
    Quaternion rotation  
)
```

Parameters

value

The vector to rotate.

rotation

The [Quaternion](#) rotation to apply.

Return Value

Returns a new [Vector2](#) containing the result of the rotation.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2, Quaternion, Vector2)

Transforms a [Vector2](#), or the vector normal (x, y, 0, 0), by a specified [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector2 value,  
    ref Quaternion rotation,  
    out Vector2 result  
)
```

Parameters

value

The vector to rotate.

rotation

The [Quaternion](#) rotation to apply.

result

[[OutAttribute](#)] An existing [Vector2](#) filled in with the result of the rotation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2[], Int32, Matrix, Vector2[], Int32, Int32)

Transforms a specified range in an array of [Vector2s](#) by a specified [Matrix](#) and places the results in a specified range in a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (
    Vector2[] sourceArray,
    int sourceIndex,
    ref Matrix matrix,
    Vector2[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The source array.

sourceIndex

The index of the first [Vector2](#) to transform in the source array.

matrix

The [Matrix](#) to transform by.

destinationArray

The destination array into which the resulting [Vector2s](#) will be written.

destinationIndex

The index of the position in the destination array where the first result [Vector2](#) should be written.

length

The number of [Vector2s](#) to be transformed.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2[], Int32, Quaternion, Vector2[], Int32, Int32)

Transforms a specified range in an array of [Vector2s](#) by a specified [Quaternion](#) and places the results in a specified range in a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (
    Vector2[] sourceArray,
    int sourceIndex,
    ref Quaternion rotation,
    Vector2[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The source array.

sourceIndex

The index of the first [Vector2](#) to transform in the source array.

rotation

The [Quaternion](#) rotation to apply.

destinationArray

The destination array into which the resulting [Vector2s](#) are written.

destinationIndex

The index of the position in the destination array where the first result [Vector2](#) should be written.

length

The number of [Vector2s](#) to be transformed.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2[], Matrix, Vector2[])

Transforms an array of [Vector2s](#) by a specified [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    Vector2[] sourceArray,  
    ref Matrix matrix,  
    Vector2[] destinationArray  
)
```

Parameters

sourceArray

The array of [Vector2s](#) to transform.

matrix

The transform [Matrix](#) to apply.

destinationArray

An existing array into which the transformed [Vector2s](#) are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Transform Method (Vector2[], Quaternion, Vector2[])

Transforms an array of [Vector2s](#) by a specified [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    Vector2[] sourceArray,  
    ref Quaternion rotation,  
    Vector2[] destinationArray  
)
```

Parameters

sourceArray

The array of [Vector2s](#) to transform.

rotation

The transform [Matrix](#) to use.

destinationArray

An existing array into which the transformed [Vector2s](#) are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.TransformNormal Method

Transforms a vector normal or array of vector normals by a matrix.

Overload List

Name	Description
Vector2.TransformNormal (Vector2, Matrix)	Transforms a 2D vector normal by a matrix.
Vector2.TransformNormal (Vector2, Matrix, Vector2)	Transforms a vector normal by a matrix.
Vector2.TransformNormal (Vector2[], Int32, Matrix, Vector2[], Int32, Int32)	Transforms a specified range in an array of Vector2 vector normals by a specified Matrix and places the results in a specified range in a destination array.
Vector2.TransformNormal (Vector2[], Matrix, Vector2[])	Transforms an array of Vector2 vector normals by a specified Matrix .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.TransformNormal Method (Vector2, Matrix)

Transforms a 2D vector normal by a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 TransformNormal (  
    Vector2 normal,  
    Matrix matrix  
)
```

Parameters

normal

The source vector.

matrix

The transformation matrix.

Return Value

The transformed normal.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.TransformNormal Method (Vector2, Matrix, Vector2)

Transforms a vector normal by a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void TransformNormal (  
    ref Vector2 normal,  
    ref Matrix matrix,  
    out Vector2 result  
)
```

Parameters

normal

The source vector.

matrix

The transformation matrix.

result

[[OutAttribute](#)] The [Vector2](#) resulting from the transformation.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.TransformNormal Method (Vector2[], Int32, Matrix, Vector2[], Int32, Int32)

Transforms a specified range in an array of [Vector2](#) vector normals by a specified [Matrix](#) and places the results in a specified range in a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void TransformNormal (
    Vector2[] sourceArray,
    int sourceIndex,
    ref Matrix matrix,
    Vector2[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The source array.

sourceIndex

The index of the first [Vector2](#) to transform in the source array.

matrix

The [Matrix](#) to apply.

destinationArray

The destination array into which the resulting [Vector2s](#) are written.

destinationIndex

The index of the position in the destination array where the first result [Vector2](#) should be written.

length

The number of vector normals to be transformed.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.TransformNormal Method (Vector2[], Matrix, Vector2[])

Transforms an array of [Vector2](#) vector normals by a specified [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void TransformNormal (  
    Vector2[] sourceArray,  
    ref Matrix matrix,  
    Vector2[] destinationArray  
)
```

Parameters

sourceArray

The array of vector normals to transform.

matrix

The transform [Matrix](#) to apply.

destinationArray

An existing array into which the transformed vector normals are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector2 Structure](#)





[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2 Properties

Public Properties

	Name	Description
 S	One	Returns a Vector2 with both of its components set to one.
 S	UnitX	Returns the unit vector for the x-axis.
 S	UnitY	Returns the unit vector for the y-axis.
 S	Zero	Returns a Vector2 with all of its components set to zero.

See Also

Reference

[Vector2 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector2.One Property

Returns a [Vector2](#) with both of its components set to one.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 One { get; }
```

Property Value

[Vector2](#) with both of its components set to one.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.UnitX Property

Returns the unit vector for the x-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 UnitX { get; }
```

Property Value

The unit vector for the x-axis.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Unity Property

Returns the unit vector for the y-axis.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Unity { get; }
```

Property Value

The unit vector for the y-axis.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector2.Zero Property

Returns a [Vector2](#) with all of its components set to zero.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector2 Zero { get; }
```

Property Value

A [Vector2](#) with all of its components set to zero.

See Also

Reference

[Vector2 Structure](#)

[Vector2 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3 Structure

Defines a vector with three components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.Vector3Converter)")]  
[SerializableAttribute]  
public struct Vector3 : IEquatable<Vector3>
```

See Also

Reference

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




Vector3 Members

The following tables list the members exposed by the Vector3 type.












Public Constructors

	Name	Description
	Vector3	Overloaded. Initializes a new instance of Vector3 .












Public Fields

	Name	Description
	X	Gets or sets the x-component of the vector.
	Y	Gets or sets the y-component of the vector.
	Z	Gets or sets the z-component of the vector.


Public Properties

	Name	Description
	Backward	Returns a unit Vector3 designating backward in a right-handed coordinate system (0, 0, 1).
	Down	Returns a unit Vector3 designating down (0, -1, 0).
	Forward	Returns a unit Vector3 designating forward in a right-handed coordinate system(0, 0, -1).
	Left	Returns a unit Vector3 designating left (-1, 0, 0).
	One	Returns a Vector3 with ones in all of its components.
	Right	Returns a unit Vector3 pointing to the right (1, 0, 0).
	UnitX	Returns the x unit Vector3 (1, 0, 0).
	UnitY	Returns the y unit Vector3 (0, 1, 0).
	UnitZ	Returns the z unit Vector3 (0, 0, 1).
	Up	Returns a unit vector designating up (0, 1, 0).
	Zero	Returns a Vector3 with all of its components set to zero.



Public Methods

	Name	Description
	Add	Overloaded. Adds two vectors.
	Barycentric	Overloaded. Returns a Vector3 containing the 3D Cartesian coordinates of a point specified in Barycentric coordinates relative to a 3D triangle.
	CatmullRom	Overloaded. Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Overloaded. Restricts a value to be within a specified range.
	Cross	Overloaded. Calculates the cross product of two vectors.
	Distance	Overloaded. Calculates the distance between two vectors.
	DistanceSquared	Overloaded. Calculates the distance between two vectors squared.
	Divide	Overloaded. Divide a vector by a scalar or another vector.
	Dot	Overloaded. Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of the vector object.

GetTy pe	(Inherited from Object .)
Herm ite	Overloaded. Performs a Hermite spline interpolation.
Lengt h	Calculates the length of the vector.
Lengt hSqu ared	Calculates the length of the vector squared.
Lerp	Overloaded. Performs a linear interpolation between two vectors.
Max	Overloaded. Returns a vector that contains the highest value from each matching pair of components.
Min	Overloaded. Returns a vector that contains the lowest value from each matching pair of components.
Multi ply	Overloaded. Multiplies a vector by a scalar or another vector.
Nega te	Overloaded. Returns a vector pointing in the opposite direction.
Norm alize	Overloaded. Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.
op_A dditio n	Adds two vectors.
op_Di vision	Overloaded. Divides a vector by a scalar or another vector.
op_E qualit y	Tests vectors for equality.
op_In equal ity	Tests vectors for inequality.
op_M ultipl y	Overloaded. Multiplies a vector by a scalar or another vector.
op_S ubtra ction	Subtracts a vector from a vector.
op_U nary Nega tion	Returns a vector pointing in the opposite direction.
Refer enceE quals	(Inherited from Object .)
Refle ct	Overloaded. Returns the reflection of a vector off a surface that has the specified normal.
Smoo thSte p	Overloaded. Interpolates between two values using a cubic equation.
Subtr act	Overloaded. Subtracts a vector from a vector.
ToStri ng	Retrieves a string representation of the current object.
Trans form	Overloaded. Transforms a Vector3 or array of Vector3 s by a specified Matrix or Quaternion .

 TransformNormal	Overloaded. Transforms a vector normal by a matrix.
---	---

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also




Reference

[Vector3 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3 Fields

Public Fields

	Name	Description
	X	Gets or sets the x-component of the vector.
	Y	Gets or sets the y-component of the vector.
	Z	Gets or sets the z-component of the vector.

See Also

Reference

[Vector3 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.X Field

Gets or sets the x-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float X
```

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Y Field

Gets or sets the y-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Y
```

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Z Field

Gets or sets the z-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Z
```

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3 Constructor

Initializes a new instance of [Vector3](#).

Overload List

Name	Description
Vector3 (Single)	Creates a new instance of Vector3 .
Vector3 (Single, Single, Single)	Initializes a new instance of Vector3 .
Vector3 (Vector2, Single)	Initializes a new instance of Vector3 .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3 Constructor (Single)

Creates a new instance of [Vector3](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 (  
    float value  
)
```

Parameters

value

Value to initialize each component to.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3 Constructor (Single, Single, Single)

Initializes a new instance of [Vector3](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 (  
    float x,  
    float y,  
    float z  
)
```

Parameters

x

Initial value for the x-component of the vector.

y

Initial value for the y-component of the vector.

z

Initial value for the z-component of the vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3 Constructor (Vector2, Single)

Initializes a new instance of [Vector3](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 (  
    Vector2 value,  
    float z  
)
```

Parameters

value

A vector containing the values to initialize x and y components with.

z

Initial value for the z-component of the vector.

See Also

Reference

[Vector3 Structure](#)























[Vector3 Members](#)














[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Vector3 Methods

Public Methods

	Name	Description
	Add	Overloaded. Adds two vectors.
	Barycentric	Overloaded. Returns a Vector3 containing the 3D Cartesian coordinates of a point specified in Barycentric coordinates relative to a 3D triangle.
	CatmullRom	Overloaded. Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Overloaded. Restricts a value to be within a specified range.
	Cross	Overloaded. Calculates the cross product of two vectors.
	Distance	Overloaded. Calculates the distance between two vectors.
	DistanceSquared	Overloaded. Calculates the distance between two vectors squared.
	Divide	Overloaded. Divide a vector by a scalar or another vector.
	Dot	Overloaded. Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of the vector object.
	GetType	(Inherited from Object .)
	Hermite	Overloaded. Performs a Hermite spline interpolation.
	Length	Calculates the length of the vector.
	LengthSquared	Calculates the length of the vector squared.
	Lerp	Overloaded. Performs a linear interpolation between two vectors.
	Max	Overloaded. Returns a vector that contains the highest value from each matching pair of components.
	Min	Overloaded. Returns a vector that contains the lowest value from each matching pair of components.
	Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
	Negate	Overloaded. Returns a vector pointing in the opposite direction.
	Normalize	Overloaded. Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.
	op_Addition	Adds two vectors.

 op_Division	Overloaded. Divides a vector by a scalar or another vector.
 op_Equality	Tests vectors for equality.
 op_Inequality	Tests vectors for inequality.
 op_Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
 op_Subtraction	Subtracts a vector from a vector.
 UnaryNegation	Returns a vector pointing in the opposite direction.
 ReferenceEquals	(Inherited from Object .)
 Reflect	Overloaded. Returns the reflection of a vector off a surface that has the specified normal.
 SmoothStep	Overloaded. Interpolates between two values using a cubic equation.
 Subtract	Overloaded. Subtracts a vector from a vector.
 ToString	Retrieves a string representation of the current object.
 Transform	Overloaded. Transforms a Vector3 or array of Vector3 s by a specified Matrix or Quaternion .
 TransformNormal	Overloaded. Transforms a vector normal by a matrix.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Vector3 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Add Method

Adds two vectors.

Overload List

Name	Description
Vector3.Add (Vector3, Vector3)	Adds two vectors.
Vector3.Add (Vector3, Vector3, Vector3)	Adds two vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Add Method (Vector3, Vector3)

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Add (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Sum of the source vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Add Method (Vector3, Vector3, Vector3)

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Add (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] Sum of the source vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Barycentric Method

Returns a [Vector3](#) containing the 3D Cartesian coordinates of a point specified in Barycentric coordinates relative to a 3D triangle.

Overload List

Name	Description
Vector3.Barycentric (Vector3, Vector3, Vector3, Single, Single)	Returns a Vector3 containing the 3D Cartesian coordinates of a point specified in Barycentric coordinates relative to a 3D triangle.
Vector3.Barycentric (Vector3, Vector3, Vector3, Single, Single, Vector3)	Returns a Vector3 containing the 3D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 3D triangle.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Barycentric Method (Vector3, Vector3, Vector3, Single, Single)

Returns a [Vector3](#) containing the 3D Cartesian coordinates of a point specified in Barycentric coordinates relative to a 3D triangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Barycentric (
    Vector3 value1,
    Vector3 value2,
    Vector3 value3,
    float amount1,
    float amount2
)
```

Parameters

value1

A [Vector3](#) containing the 3D Cartesian coordinates of vertex 1 of the triangle.

value2

A [Vector3](#) containing the 3D Cartesian coordinates of vertex 2 of the triangle.

value3

A [Vector3](#) containing the 3D Cartesian coordinates of vertex 3 of the triangle.

amount1

Barycentric coordinate b_2 , which expresses the weighting factor toward vertex 2 (specified in *value2*).

amount2

Barycentric coordinate b_3 , which expresses the weighting factor toward vertex 3 (specified in *value3*).

Return Value

A new [Vector3](#) containing the 3D Cartesian coordinates of the specified point.

Remarks

About Barycentric Coordinates

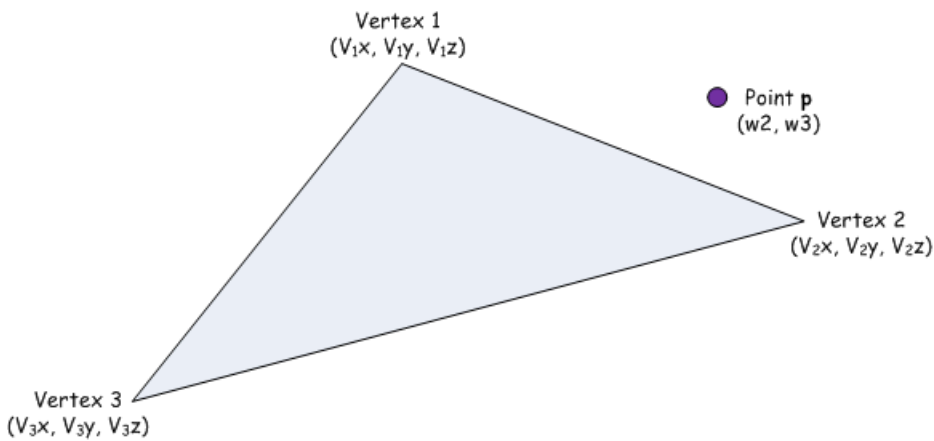
Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by 3 weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas:

$$\begin{aligned} P_x &= (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\ P_z &= (b_1 * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \end{aligned}$$

Such triple weighting factors **b1**, **b2**, and **b3** are called *Barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that $(k * b_1)$, $(k * b_2)$, and $(k * b_3)$ are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of Barycentric coordinates is normalized so that $b_1 + b_2 + b_3 = 1$, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** equals $(1 - b_2 - b_3)$.



What Vector3 Barycentric Does

The Vector3 [Barycentric](#) method takes three vectors specifying the Cartesian coordinates of the triangle vertices, **V1**, **V2**, and **V3**, and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

[Barycentric](#) then calculates the Cartesian coordinate of **P** as follows:

$$\begin{aligned}
 P_x &= ((1 - b_2 - b_3) * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}) ; \\
 P_y &= ((1 - b_2 - b_3) * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}) ; \\
 P_z &= ((1 - b_2 - b_3) * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}) ;
 \end{aligned}$$

Thus, to calculate the 3D Cartesian coordinates of **P**, you would pass the coordinates of the triangle vertices to [Barycentric](#) together with the appropriate normalized Barycentric (areal) coordinates of **P**.

The following relationships may be useful:

- If (*amount1* <= 0) and (*amount2* >= 0) and (1 - *amount1* - *amount2* >= 0), then the point is inside the triangle defined by *value1*, *value2*, and *value3*.
- If (*amount1* == 0) and (*amount2* >= 0) and (1 - *amount1* - *amount2* >= 0), then the point is on the line defined by *value1* and *value3*.
- If (*amount1* >= 0) and (*amount2* == 0) and (1 - *amount1* - *amount2* >= 0), then the point is on the line defined by *value1* and *value2*.
- If (*amount1* >= 0) and (*amount2* >= 0) and (1 - *amount1* - *amount2* == 0), then the point is on the line defined by *value2* and *value3*.

Barycentric coordinates are a form of general coordinates. In this context, using barycentric coordinates represents a change in coordinate systems. What holds true for Cartesian coordinates holds true for barycentric coordinates.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Barycentric Method (Vector3, Vector3, Vector3, Single, Single, Vector3)

Returns a [Vector3](#) containing the 3D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 3D triangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Barycentric (
    ref Vector3 value1,
    ref Vector3 value2,
    ref Vector3 value3,
    float amount1,
    float amount2,
    out Vector3 result
)
```

Parameters

value1

A [Vector3](#) containing the 3D Cartesian coordinates of vertex 1 of the triangle.

value2

A [Vector3](#) containing the 3D Cartesian coordinates of vertex 2 of the triangle.

value3

A [Vector3](#) containing the 3D Cartesian coordinates of vertex 3 of the triangle.

amount1

Barycentric coordinate **b2**, which expresses the weighting factor toward vertex 2 (specified in *value2*).

amount2

Barycentric coordinate **b3**, which expresses the weighting factor toward vertex 3 (specified in *value3*).

result

[[OutAttribute](#)] The 3D Cartesian coordinates of the specified point are placed in this [Vector3](#) on exit.

Remarks

About Barycentric Coordinates

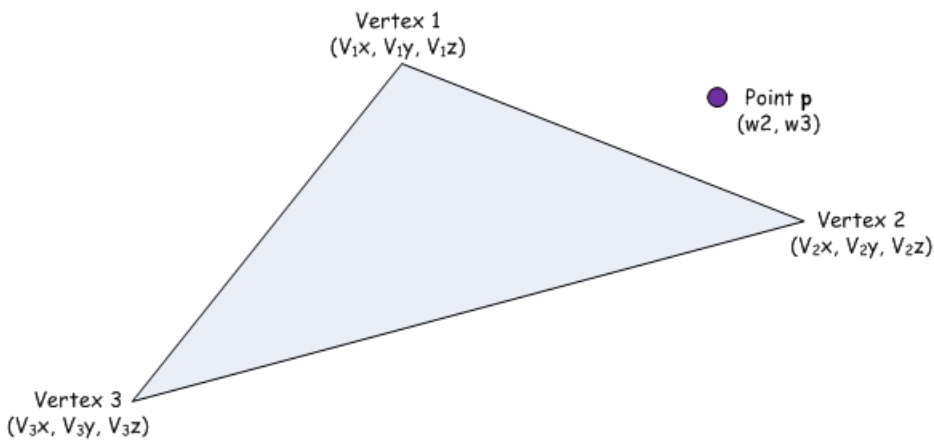
Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by three weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas:

$$\begin{aligned} P_x &= (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\ P_z &= (b_1 * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \end{aligned}$$

Such triple weighting factors **b1**, **b2**, and **b3** are called *barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that **(k * b1)**, **(k * b2)**, and **(k * b3)** are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of barycentric coordinates is normalized so that **b1 + b2 + b3 = 1**, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** equals **(1 - b2 - b3)**.



What Vector3 Barycentric Does

The Vector3 [Barycentric](#) method takes three vectors specifying the Cartesian coordinates of the triangle vertices, **V1**, **V2**, and **V3**, and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

[Barycentric](#) then calculates the Cartesian coordinate of **P** as follows:

$$\begin{aligned} P_x &= ((1 - b_2 - b_3) * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= ((1 - b_2 - b_3) * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\ P_z &= ((1 - b_2 - b_3) * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \end{aligned}$$

Thus, to calculate the 3D Cartesian coordinates of **P**, you would pass the coordinates of the triangle vertices to [Barycentric](#) together with the appropriate normalized Barycentric (areal) coordinates of **P**.

The following relationships may be useful.

- If $(amount1 \leq 0)$ and $(amount2 \geq 0)$ and $(1 - amount1 - amount2 \geq 0)$, then the point is inside the triangle defined by *value1*, *value2*, and *value3*.
- If $(amount1 == 0)$ and $(amount2 \geq 0)$ and $(1 - amount1 - amount2 \geq 0)$, then the point is on the line defined by *value1* and *value3*.
- If $(amount1 \geq 0)$ and $(amount2 == 0)$ and $(1 - amount1 - amount2 \geq 0)$, then the point is on the line defined by *value1* and *value2*.
- If $(amount1 \geq 0)$ and $(amount2 \geq 0)$ and $(1 - amount1 - amount2 == 0)$, then the point is on the line defined by *value2* and *value3*.

Barycentric coordinates are a form of general coordinates. In this context, using barycentric coordinates represents a change in coordinate systems. What holds true for Cartesian coordinates holds true for barycentric coordinates.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.CatmullRom Method

Performs a Catmull-Rom interpolation using the specified positions.

Overload List

Name	Description
Vector3.CatmullRom (Vector3, Vector3, Vector3, Vector3, Single)	Performs a Catmull-Rom interpolation using the specified positions.
Vector3.CatmullRom (Vector3, Vector3, Vector3, Vector3, Single, Vector3)	Performs a Catmull-Rom interpolation using the specified positions.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.CatmullRom Method (Vector3, Vector3, Vector3, Vector3, Single)

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 CatmullRom (  
    Vector3 value1,  
    Vector3 value2,  
    Vector3 value3,  
    Vector3 value4,  
    float amount  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

Return Value

A vector that is the result of the Catmull-Rom interpolation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.CatmullRom Method (Vector3, Vector3, Vector3, Vector3, Single, Vector3)

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CatmullRom (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    ref Vector3 value3,  
    ref Vector3 value4,  
    float amount,  
    out Vector3 result  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

result

[[OutAttribute](#)] A vector that is the result of the Catmull-Rom interpolation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Clamp Method

Restricts a value to be within a specified range.

Overload List

Name	Description
Vector3.Clamp (Vector3, Vector3, Vector3)	Restricts a value to be within a specified range.
Vector3.Clamp (Vector3, Vector3, Vector3, Vector3)	Restricts a value to be within a specified range.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Clamp Method (Vector3, Vector3, Vector3)

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Clamp (  
    Vector3 value1,  
    Vector3 min,  
    Vector3 max  
)
```

Parameters

value1

The value to clamp.

min

The minimum value.

max

The maximum value.

Return Value

The clamped value.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Clamp Method (Vector3, Vector3, Vector3, Vector3)

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Clamp (  
    ref Vector3 value1,  
    ref Vector3 min,  
    ref Vector3 max,  
    out Vector3 result  
)
```

Parameters

value1

The value to clamp.

min

The minimum value.

max

The maximum value.

result

[[OutAttribute](#)] The clamped value.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Cross Method

Calculates the cross product of two vectors.

Overload List

Name	Description
Vector3.Cross (Vector3, Vector3)	Calculates the cross product of two vectors.
Vector3.Cross (Vector3, Vector3, Vector3)	Calculates the cross product of two vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Cross Method (Vector3, Vector3)

Calculates the cross product of two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Cross (  
    Vector3 vector1,  
    Vector3 vector2  
)
```

Parameters

vector1

Source vector.

vector2

Source vector.

Return Value

Cross product of the source vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Cross Method (Vector3, Vector3, Vector3)

Calculates the cross product of two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Cross (  
    ref Vector3 vector1,  
    ref Vector3 vector2,  
    out Vector3 result  
)
```

Parameters

vector1

Source vector.

vector2

Source vector.

result

[[OutAttribute](#)] The cross product of the vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Distance Method

Calculates the distance between two vectors.

Overload List

Name	Description
Vector3.Distance (Vector3, Vector3)	Calculates the distance between two vectors.
Vector3.Distance (Vector3, Vector3, Single)	Calculates the distance between two vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Distance Method (Vector3, Vector3)

Calculates the distance between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Distance (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Distance between the source vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Distance Method (Vector3, Vector3, Single)

Calculates the distance between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Distance (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out float result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The distance between the vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.DistanceSquared Method

Calculates the distance between two vectors squared.

Overload List

Name	Description
Vector3.DistanceSquared (Vector3, Vector3)	Calculates the distance between two vectors squared.
Vector3.DistanceSquared (Vector3, Vector3, Single)	Calculates the distance between two vectors squared.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.DistanceSquared Method (Vector3, Vector3)

Calculates the distance between two vectors squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float DistanceSquared (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The distance between the source vectors squared.

Remarks

Distance is calculated with the following formula:

$$\sqrt{(a - b)^2}$$

Distance squared is the value before taking the square root. Distance squared can often be used in place of distance if relative comparisons are being made. For example, consider three points A, B, and C. To determine whether B or C is further from A, compare the distance between A and B to the distance between A and C. Calculating the two distances involves two square roots, which are computationally expensive. However, using distance squared provides the same information and avoids calculating two square roots.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.DistanceSquared Method (Vector3, Vector3, Single)

Calculates the distance between two vectors squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void DistanceSquared (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out float result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The distance between the two vectors squared.

Remarks

Distance is calculated with the formula:

$$\sqrt{(a - b)^2}$$

Distance squared is the value before taking the square root. Distance squared can often be used in place of distance if relative comparisons are being made. For example, consider three points A, B, and C. To determine whether B or C is further from A, compare the distance between A and B to the distance between A and C. Calculating the two distances involves two square roots, which are computationally expensive. However, using distance squared provides the same information and avoids calculating two square roots.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Divide Method

Divide a vector by a scalar or another vector.

Overload List

Name	Description
Vector3.Divide (Vector3, Single)	Divides a vector by a scalar value.
Vector3.Divide (Vector3, Single, Vector3)	Divides a vector by a scalar value.
Vector3.Divide (Vector3, Vector3)	Divides the components of a vector by the components of another vector.
Vector3.Divide (Vector3, Vector3, Vector3)	Divides the components of a vector by the components of another vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Divide Method (Vector3, Single)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Divide (  
    Vector3 value1,  
    float value2  
)
```

Parameters

value1

Source vector.

value2

The divisor.

Return Value

The source vector divided by *b*.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Divide Method (Vector3, Single, Vector3)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Vector3 value1,  
    float value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

The divisor.

result

[[OutAttribute](#)] The result of the division.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Divide Method (Vector3, Vector3)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Divide (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Divisor vector.

Return Value

The result of dividing the vectors.

Remarks

Division of a vector by another vector is not mathematically defined. This method simply divides each component of *a* by the matching component of *b*.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Divide Method (Vector3, Vector3, Vector3)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

The divisor.

result

[[OutAttribute](#)] The result of the division.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Dot Method

Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Overload List

Name	Description
Vector3.Dot (Vector3, Vector3)	Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.
Vector3.Dot (Vector3, Vector3, Single)	Calculates the dot product of two vectors and writes the result to a user-specified variable. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Remarks

The dot product is also known as the inner product.

For any two vectors, the dot product is defined as:

```
Vector3.Dot( vector1, vector2 ) == (vector1.X * vector2.X) + (vector1.Y * vector2.Y) + (vector1.Z * vector2.Z)
```

The result of this calculation, plus or minus some margin to account for floating point error, is equal to:

```
Length( vector1 ) * Length( vector2 ) * System.Math.Cos( theta )
```

Here, *theta* is the angle between the two vectors.

If *vector1* and *vector2* are unit vectors, the length of each vector will be equal to 1. So, when *vector1* and *vector2* are unit vectors, the dot product is simply equal to the cosine of the angle between the two vectors. The following calculation:

```
Vector3.Dot( vector1.Normalize(), vector2.Normalize() )
```

Is equivalent to:

```
System.Math.Cos( theta )
```

Therefore, if *vector1* and *vector2* are unit vectors (i.e., we've called [Normalize](#) on each vector to make it a unit vector), without knowing the value of *theta* or using a potentially processor-intensive trigonometric function, the dot product can tell us the following things:

- If `Vector3.Dot(vector1.Normalize(), vector2.Normalize()) > 0`, the angle between the two vectors is less than 90 degrees.
- If `Vector3.Dot(vector1.Normalize(), vector2.Normalize()) < 0`, the angle between the two vectors is more than 90 degrees.
- If `Vector3.Dot(vector1.Normalize(), vector2.Normalize()) == 0`, the angle between the two vectors is 90 degrees; that is, the vectors are orthogonal.
- If `Vector3.Dot(vector1.Normalize(), vector2.Normalize()) == 1`, the angle between the two vectors is 0 degrees; that is, the vectors point in the same direction and are parallel.
- If `Vector3.Dot(vector1.Normalize(), vector2.Normalize()) == -1`, the angle between the two vectors is 180 degrees; that is, the vectors point in opposite directions and are parallel.

⚠Caution

Because of floating point error, two orthogonal vectors may not return a dot product that is exactly zero. It might be zero plus some amount of floating point error. In your code, you will want to determine what amount of error is acceptable in your calculation, and take that into account when you do your comparisons.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Dot Method (Vector3, Vector3)

Calculates the dot product of two vectors. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Dot (
    Vector3 vector1,
    Vector3 vector2
)
```

Parameters

vector1

Source vector.

vector2

Source vector.

Return Value

The dot product of the two vectors.

Remarks

The dot product is also known as the inner product.

For any two vectors, the dot product is defined as:

$$\text{Vector3.Dot}(\text{vector1}, \text{vector2}) == (\text{vector1.X} * \text{vector2.X}) + (\text{vector1.Y} * \text{vector2.Y}) + (\text{vector1.Z} * \text{vector2.Z})$$

The result of this calculation, plus or minus some margin to account for floating point error, is equal to:

$$\text{Length}(\text{vector1}) * \text{Length}(\text{vector2}) * \text{System.Math.Cos}(\text{theta})$$

Here, *theta* is the angle between the two vectors.

If *vector1* and *vector2* are unit vectors, the length of each vector will be equal to 1. So, when *vector1* and *vector2* are unit vectors, the dot product is simply equal to the cosine of the angle between the two vectors. The following calculation:

$$\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}())$$

Is equivalent to:

$$\text{System.Math.Cos}(\text{theta})$$

Therefore, if *vector1* and *vector2* are unit vectors (i.e., we've called [Normalize](#) on each vector to make it a unit vector), without knowing the value of *theta* or using a potentially processor-intensive trigonometric function, the dot product can tell us the following things:

- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) > 0$, the angle between the two vectors is less than 90 degrees.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) < 0$, the angle between the two vectors is more than 90 degrees.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == 0$, the angle between the two vectors is 90 degrees; that is, the vectors are orthogonal.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == 1$, the angle between the two vectors is 0 degrees; that is, the vectors point in the same direction and are parallel.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == -1$, the angle between the two vectors is 180 degrees; that is, the vectors point in opposite directions and are parallel.

⚠ Caution

Because of floating point error, two orthogonal vectors may not return a dot product that is exactly zero. It might be zero plus some amount of floating point error. In your code, you will want to determine what amount of error is acceptable in your calculation, and take that into account when you do your comparisons.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Dot Method (Vector3, Vector3, Single)

Calculates the dot product of two vectors and writes the result to a user-specified variable. If the two vectors are unit vectors, the dot product returns a floating point value between -1 and 1 that can be used to determine some properties of the angle between two vectors. For example, it can show whether the vectors are orthogonal, parallel, or have an acute or obtuse angle between them.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Dot (
    ref Vector3 vector1,
    ref Vector3 vector2,
    out float result
)
```

Parameters

vector1

Source vector.

vector2

Source vector.

result

[[OutAttribute](#)] The dot product of the two vectors.

Remarks

The dot product is also known as the inner product.

For any two vectors, the dot product is defined as:

$$\text{Vector3.Dot}(\text{vector1}, \text{vector2}) == (\text{vector1.X} * \text{vector2.X}) + (\text{vector1.Y} * \text{vector2.Y}) + (\text{vector1.Z} * \text{vector2.Z})$$

The result of this calculation, plus or minus some margin to account for floating point error, is equal to:

$$\text{Length}(\text{vector1}) * \text{Length}(\text{vector2}) * \text{System.Math.Cos}(\text{theta})$$

Here, *theta* is the angle between the two vectors.

If *vector1* and *vector2* are unit vectors, the length of each vector will be equal to 1. So, when *vector1* and *vector2* are unit vectors, the dot product is simply equal to the cosine of the angle between the two vectors. The following calculation:

$$\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}())$$

Is equivalent to:

$$\text{System.Math.Cos}(\text{theta})$$

Therefore, if *vector1* and *vector2* are unit vectors (i.e., we've called [Normalize](#) on each vector to make it a unit vector), without knowing the value of *theta* or using a potentially processor-intensive trigonometric function, the dot product can tell us the following things:

- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) > 0$, the angle between the two vectors is less than 90 degrees.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) < 0$, the angle between the two vectors is more than 90 degrees.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == 0$, the angle between the two vectors is 90 degrees; that is, the vectors are orthogonal.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == 1$, the angle between the two vectors is 0 degrees; that is, the vectors point in the same direction and are parallel.
- If $\text{Vector3.Dot}(\text{vector1.Normalize}(), \text{vector2.Normalize}()) == -1$, the angle between the two vectors is 180 degrees; that is, the vectors point in opposite directions and are parallel.

⚠ Caution

Because of floating point error, two orthogonal vectors may not return a dot product that is exactly zero. It might be zero plus some amount of floating point error. In your code, you will want to determine what amount of error is acceptable in your calculation, and take that into account when you do your comparisons.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Vector3.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Vector3.Equals (Vector3)	Determines whether the specified Object is equal to the Vector3 .
Vector3.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object to make the comparison with.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Equals Method (Vector3)

Determines whether the specified [Object](#) is equal to the [Vector3](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Vector3 other  
)
```

Parameters

other

The [Vector3](#) to compare with the current [Vector3](#).

Return Value

true if the specified [Vector3](#) is equal to the current [Vector3](#); **false** otherwise.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.GetHashCode Method

Gets the hash code of the vector object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code of the vector object.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Hermite Method

Performs a Hermite spline interpolation.

Overload List

Name	Description
Vector3.Hermite (Vector3, Vector3, Vector3, Vector3, Single)	Performs a Hermite spline interpolation.
Vector3.Hermite (Vector3, Vector3, Vector3, Vector3, Single, Vector3)	Performs a Hermite spline interpolation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Hermite Method (Vector3, Vector3, Vector3, Vector3, Single)

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Hermite (  
    Vector3 value1,  
    Vector3 tangent1,  
    Vector3 value2,  
    Vector3 tangent2,  
    float amount  
)
```

Parameters

value1

Source position vector.

tangent1

Source tangent vector.

value2

Source position vector.

tangent2

Source tangent vector.

amount

Weighting factor.

Return Value

The result of the Hermite spline interpolation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Hermite Method (Vector3, Vector3, Vector3, Vector3, Single, Vector3)

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Hermite (  
    ref Vector3 value1,  
    ref Vector3 tangent1,  
    ref Vector3 value2,  
    ref Vector3 tangent2,  
    float amount,  
    out Vector3 result  
)
```

Parameters

value1

Source position vector.

tangent1

Source tangent vector.

value2

Source position vector.

tangent2

Source tangent vector.

amount

Weighting factor.

result

[[OutAttribute](#)] The result of the Hermite spline interpolation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Length Method

Calculates the length of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Length ()
```

Return Value

The length of the vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.LengthSquared Method

Calculates the length of the vector squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float LengthSquared ()
```

Return Value

The length of the vector squared.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Lerp Method

Performs a linear interpolation between two vectors.

Overload List

Name	Description
Vector3.Lerp (Vector3, Vector3, Single)	Performs a linear interpolation between two vectors.
Vector3.Lerp (Vector3, Vector3, Single, Vector3)	Performs a linear interpolation between two vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Lerp Method (Vector3, Vector3, Single)

Performs a linear interpolation between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Lerp (  
    Vector3 value1,  
    Vector3 value2,  
    float amount  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Value between 0 and 1 indicating the weight of *value2*.

Return Value

The linear interpolation of the two vectors.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned; a value of 1 will cause *value2* to be returned.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Lerp Method (Vector3, Vector3, Single, Vector3)

Performs a linear interpolation between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Lerp (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    float amount,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Value between 0 and 1 indicating the weight of *value2*.

result

[[OutAttribute](#)] The result of the interpolation.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned; a value of 1 will cause *value2* to be returned.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Max Method

Returns a vector that contains the highest value from each matching pair of components.

Overload List

Name	Description
Vector3.Max (Vector3, Vector3)	Returns a vector that contains the highest value from each matching pair of components.
Vector3.Max (Vector3, Vector3, Vector3)	Returns a vector that contains the highest value from each matching pair of components.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Max Method (Vector3, Vector3)

Returns a vector that contains the highest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Max (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The maximized vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Max Method (Vector3, Vector3, Vector3)

Returns a vector that contains the highest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Max (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The maximized vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Min Method

Returns a vector that contains the lowest value from each matching pair of components.

Overload List

Name	Description
Vector3.Min (Vector3, Vector3)	Returns a vector that contains the lowest value from each matching pair of component s.
Vector3.Min (Vector3, Vector3, Vector3)	Returns a vector that contains the lowest value from each matching pair of component s.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Min Method (Vector3, Vector3)

Returns a vector that contains the lowest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Min (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The minimized vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Min Method (Vector3, Vector3, Vector3)

Returns a vector that contains the lowest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Min (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The minimized vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Multiply Method

Multiplies a vector by a scalar or another vector.

Overload List

Name	Description
Vector3.Multiply (Vector3, Single)	Multiplies a vector by a scalar value.
Vector3.Multiply (Vector3, Single, Vector3)	Multiplies a vector by a scalar value.
Vector3.Multiply (Vector3, Vector3)	Multiplies the components of two vectors by each other.
Vector3.Multiply (Vector3, Vector3, Vector3)	Multiplies the components of two vectors by each other.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Multiply Method (Vector3, Single)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Multiply (  
    Vector3 value1,  
    float scaleFactor  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Multiply Method (Vector3, Single, Vector3)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Vector3 value1,  
    float scaleFactor,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Multiply Method (Vector3, Vector3)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Multiply (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the multiplication.

Remarks

Multiplication performed by this method is not vector multiplication (dot product and cross product) but multiplication of the corresponding components of each vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Multiply Method (Vector3, Vector3, Vector3)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Negate Method

Returns a vector pointing in the opposite direction.

Overload List

Name	Description
Vector3.Negate (Vector3)	Returns a vector pointing in the opposite direction.
Vector3.Negate (Vector3, Vector3)	Returns a vector pointing in the opposite direction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Negate Method (Vector3)

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Negate (  
    Vector3 value  
)
```

Parameters

value

Source vector.

Return Value

Vector pointing in the opposite direction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Negate Method (Vector3, Vector3)

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Negate (  
    ref Vector3 value,  
    out Vector3 result  
)
```

Parameters

value

Source vector.

result

[[OutAttribute](#)] Vector pointing in the opposite direction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Normalize Method

Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.

Overload List

Name	Description
Vector3.Normalize ()	Turns the current vector into a unit vector. The result is a vector one unit in length pointing in the same direction as the original vector.
Vector3.Normalize (Vector3)	Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.
Vector3.Normalize (Vector3, Vector3)	Creates a unit vector from the specified vector, writing the result to a user-specified variable. The result is a vector one unit in length pointing in the same direction as the original vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Normalize Method ()

Turns the current vector into a unit vector. The result is a vector one unit in length pointing in the same direction as the original vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Normalize ()
```

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Normalize Method (Vector3)

Creates a unit vector from the specified vector. The result is a vector one unit in length pointing in the same direction as the original vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Normalize (  
    Vector3 value  
)
```

Parameters

value

The source [Vector3](#).

Return Value

The created unit vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Normalize Method (Vector3, Vector3)

Creates a unit vector from the specified vector, writing the result to a user-specified variable. The result is a vector one unit in length pointing in the same direction as the original vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Normalize (  
    ref Vector3 value,  
    out Vector3 result  
)
```

Parameters

value

Source vector.

result

[[OutAttribute](#)] The normalized vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Addition Method

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Addition (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Sum of the vectors.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Division Method

Divides a vector by a scalar or another vector.

Overload List

Name	Description
Vector3.op_Division (Vector3, Single)	Divides a vector by a scalar value.
Vector3.op_Division (Vector3, Vector3)	Divides the components of a vector by the components of another vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.op_Division Method (Vector3, Single)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Division (  
    Vector3 value,  
    float divider  
)
```

Parameters

value

Source vector.

divider

The divisor.

Return Value

The source vector divided by *div*.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Division Method (Vector3, Vector3)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Division (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Divisor vector.

Return Value

The result of dividing the vectors.

Remarks

Division of a vector by another vector is not mathematically defined. This method simply divides each component of *a* by the matching component of *b*.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Equality Method

Tests vectors for equality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

true if the vectors are equal; **false** otherwise.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Inequality Method

Tests vectors for inequality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Vector to compare.

value2

Vector to compare.

Return Value

true if the vectors are not equal; **false** otherwise.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Multiply Method

Multiplies a vector by a scalar or another vector.

Overload List

Name	Description
Vector3.op_Multiply (Single, Vector3)	Multiplies a vector by a scalar value.
Vector3.op_Multiply (Vector3, Single)	Multiplies a vector by a scalar value.
Vector3.op_Multiply (Vector3, Vector3)	Multiplies the components of two vectors by each other.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.op_Multiply Method (Single, Vector3)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Multiply (  
    float scaleFactor,  
    Vector3 value  
)
```

Parameters

scaleFactor

Scalar value.

value

Source vector.

Return Value

Result of the multiplication.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Multiply Method (Vector3, Single)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Multiply (  
    Vector3 value,  
    float scaleFactor  
)
```

Parameters

value

Source vector.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Multiply Method (Vector3, Vector3)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Multiply (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the multiplication.

Remarks

Multiplication performed by this method is not vector multiplication (dot product and cross product) but multiplication of the corresponding components of each vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_Subtraction Method

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_Subtraction (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the subtraction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.op_UnaryNegation Method

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 op_UnaryNegation (  
    Vector3 value  
)
```

Parameters

value

Source vector.

Return Value

Vector pointing in the opposite direction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Reflect Method

Returns the reflection of a vector off a surface that has the specified normal.

Overload List

Name	Description
Vector3.Reflect (Vector3, Vector3)	Returns the reflection of a vector off a surface that has the specified normal.
Vector3.Reflect (Vector3, Vector3, Vector3)	Returns the reflection of a vector off a surface that has the specified normal.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Reflect Method (Vector3, Vector3)

Returns the reflection of a vector off a surface that has the specified normal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Reflect (
    Vector3 vector,
    Vector3 normal
)
```

Parameters

vector

Source vector.

normal

Normal of the surface.

Return Value

The reflected vector.

Remarks [Reflect](#) only gives the direction of a reflection off a surface, it does not determine whether the original vector was close enough to the surface to hit it.

Example Determining the position of a moving particle taking into account a surface being struck.

C#

```
static Nullable<float> RayPlaneIntersection(
    ref Ray ray, ref Plane plane)
{
    Vector3 PointOnPlane = plane.D * plane.Normal;
    float DotNormalDir = Vector3.Dot(plane.Normal, ray.Direction);
    if (DotNormalDir == 0) return null;
    float d = Vector3.Dot(plane.Normal, PointOnPlane - ray.Position) /
        DotNormalDir;
    return d;
}

// Set the position of the particle.
Vector3 position = new Vector3(-19, 0, -3);
// Set the direction in which the particle is moving.
Vector3 direction = new Vector3(1, 0, 0);
// Set the speed at which the particle is moving.
float speed = 4.0f;

//Define a plane with a slope of -1 with respect to the xz-axis
//and passing through (-5, 0, -5).
Plane surface = new Plane(new Vector3(-1f / (float)Math.Sqrt(2), 0,
    -1f / (float)Math.Sqrt(2)), 5f * (float)Math.Sqrt(2));

void MoveParticle()
{
    Ray ray = new Ray(position, direction);

    bool intersected = true;

    Nullable<float> collisionDistance =
        RayPlaneIntersection(ref ray, ref surface);

    // The direction of movement is parallel to the plane.
    if (collisionDistance.HasValue == false)
    {
        intersected = false;
    }
    // The intersection is opposite the direction of movement.
```

```

else if ((collisionDistance.Value <= 0 && speed > 0) ||
(collisionDistance.Value > 0 && speed <= 0))
{
    intersected = false;
}
// The intersection is further away than the distance moved.
else if (Math.Abs(collisionDistance.Value) > Math.Abs(speed))
{
    intersected = false;
}

// The particle hit the surface, so reflect it off the surface.
if (intersected == true)
{
    // If the vector position + direction * speed passes through
    // the plane, calculate the reflection.
    Vector3 reflectionVector =
        Vector3.Reflect(direction, surface.Normal);

    // Calculate the distance left to move after hitting the surface
    float lengthOfReflection = (speed - collisionDistance.Value);
    Vector3 newPosition = position + direction *
        collisionDistance.Value + lengthOfReflection *
        reflectionVector;
    position = newPosition;
    direction = reflectionVector;
}
else // There is no intersection, so just move normally.
{
    position += direction * speed;
}
}

```

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Reflect Method (Vector3, Vector3, Vector3)

Returns the reflection of a vector off a surface that has the specified normal.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Reflect (
    ref Vector3 vector,
    ref Vector3 normal,
    out Vector3 result
)
```

Parameters

vector

Source vector.

normal

Normal of the surface.

result

[OutAttribute] The reflected vector.

Remarks [Reflect](#) only gives the direction of a reflection off a surface, it does not determine whether the original vector was close enough to the surface to hit it.

Example Determining the position of a moving particle taking into account a surface being struck.

C#

```
static Nullable<float> RayPlaneIntersection(
    ref Ray ray, ref Plane plane)
{
    Vector3 PointOnPlane = plane.D * plane.Normal;
    float DotNormalDir = Vector3.Dot(plane.Normal, ray.Direction);
    if (DotNormalDir == 0) return null;
    float d = Vector3.Dot(plane.Normal, PointOnPlane - ray.Position) /
        DotNormalDir;
    return d;
}

// Set the position of the particle.
Vector3 position = new Vector3(-19, 0, -3);
// Set the direction in which the particle is moving.
Vector3 direction = new Vector3(1, 0, 0);
// Set the speed at which the particle is moving.
float speed = 4.0f;

//Define a plane with a slope of -1 with respect to the xz-axis
//and passing through (-5, 0, -5).
Plane surface = new Plane(new Vector3(-1f / (float)Math.Sqrt(2), 0,
    -1f / (float)Math.Sqrt(2)), 5f * (float)Math.Sqrt(2));

void MoveParticle()
{
    Ray ray = new Ray(position, direction);

    bool intersected = true;

    Nullable<float> collisionDistance =
        RayPlaneIntersection(ref ray, ref surface);

    // The direction of movement is parallel to the plane.
    if (collisionDistance.HasValue == false)
    {
        intersected = false;
    }
}
```

```

// The intersection is opposite the direction of movement.
else if ((collisionDistance.Value <= 0 && speed > 0) ||
(collisionDistance.Value > 0 && speed <= 0))
{
    intersected = false;
}
// The intersection is further away than the distance moved.
else if (Math.Abs(collisionDistance.Value) > Math.Abs(speed))
{
    intersected = false;
}

// The particle hit the surface, so reflect it off the surface.
if (intersected == true)
{
    // If the vector position + direction * speed passes through
    // the plane, calculate the reflection.
    Vector3 reflectionVector =
        Vector3.Reflect(direction, surface.Normal);

    // Calculate the distance left to move after hitting the surface
    float lengthOfReflection = (speed - collisionDistance.Value);
    Vector3 newPosition = position + direction *
        collisionDistance.Value + lengthOfReflection *
        reflectionVector;
    position = newPosition;
    direction = reflectionVector;
}
else // There is no intersection, so just move normally.
{
    position += direction * speed;
}
}

```

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.SmoothStep Method

Interpolates between two values using a cubic equation.

Overload List

Name	Description
Vector3.SmoothStep (Vector3, Vector3, Single)	Interpolates between two values using a cubic equation.
Vector3.SmoothStep (Vector3, Vector3, Single, Vector3)	Interpolates between two values using a cubic equation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.SmoothStep Method (Vector3, Vector3, Single)

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 SmoothStep (  
    Vector3 value1,  
    Vector3 value2,  
    float amount  
)
```

Parameters

value1

Source value.

value2

Source value.

amount

Weighting value.

Return Value

Interpolated value.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.SmoothStep Method (Vector3, Vector3, Single, Vector3)

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void SmoothStep (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    float amount,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Weighting value.

result

[[OutAttribute](#)] The interpolated value.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Subtract Method

Subtracts a vector from a vector.

Overload List

Name	Description
Vector3.Subtract (Vector3, Vector3)	Subtracts a vector from a vector.
Vector3.Subtract (Vector3, Vector3, Vector3)	Subtracts a vector from a vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Subtract Method (Vector3, Vector3)

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Subtract (  
    Vector3 value1,  
    Vector3 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the subtraction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Subtract Method (Vector3, Vector3, Vector3)

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Subtract (  
    ref Vector3 value1,  
    ref Vector3 value2,  
    out Vector3 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The result of the subtraction.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.ToString Method

Retrieves a string representation of the current object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method

Transforms a [Vector3](#) or array of [Vector3](#)s by a specified [Matrix](#) or [Quaternion](#).

Overload List

Name	Description
Vector3.Transform (Vector3, Matrix)	Transforms a 3D vector by the given matrix.
Vector3.Transform (Vector3, Matrix, Vector3)	Transforms a Vector3 by the given Matrix .
Vector3.Transform (Vector3, Quaternion)	Transforms a Vector3 by a specified Quaternion rotation.
Vector3.Transform (Vector3, Quaternion, Vector3)	Transforms a Vector3 by a specified Quaternion rotation.
Vector3.Transform (Vector3[], Int32, Matrix, Vector3[], Int32, Int32)	Applies a specified transform Matrix to a specified range of an array of Vector3 s and writes the results into a specified range of a destination array.
Vector3.Transform (Vector3[], Int32, Quaternion, Vector3[], Int32, Int32)	Applies a specified Quaternion rotation to a specified range of an array of Vector3 s and writes the results into a specified range of a destination array.
Vector3.Transform (Vector3[], Matrix, Vector3[])	Transforms a source array of Vector3 s by a specified Matrix and writes the results to an existing destination array.
Vector3.Transform (Vector3[], Quaternion, Vector3[])	Transforms a source array of Vector3 s by a specified Quaternion rotation and writes the results to an existing destination array.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Transform Method (Vector3, Matrix)

Transforms a 3D vector by the given matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Transform (  
    Vector3 position,  
    Matrix matrix  
)
```

Parameters

position

The source vector.

matrix

The transformation matrix.

Return Value

The transformed vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide

[How To: Transform a Point with a Matrix](#)

[How To: Rotate and Move a Camera](#)

[How To: Make a First-Person Camera](#)

[How To: Make a Third-Person Camera](#)

[How To: Detect Whether a User Clicked a 3D Object](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3, Matrix, Vector3)

Transforms a [Vector3](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector3 position,  
    ref Matrix matrix,  
    out Vector3 result  
)
```

Parameters

position

The source [Vector3](#).

matrix

The transformation [Matrix](#).

result

[[OutAttribute](#)] The transformed vector.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3, Quaternion)

Transforms a [Vector3](#) by a specified [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Transform (  
    Vector3 value,  
    Quaternion rotation  
)
```

Parameters

value

The [Vector3](#) to rotate.

rotation

The [Quaternion](#) rotation to apply.

Return Value

Returns a new [Vector3](#) that results from the rotation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3, Quaternion, Vector3)

Transforms a [Vector3](#) by a specified [Quaternion](#) rotation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector3 value,  
    ref Quaternion rotation,  
    out Vector3 result  
)
```

Parameters

value

The [Vector3](#) to rotate.

rotation

The [Quaternion](#) rotation to apply.

result

[[OutAttribute](#)] An existing [Vector3](#) filled in with the results of the rotation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3[], Int32, Matrix, Vector3[], Int32, Int32)

Applies a specified transform [Matrix](#) to a specified range of an array of [Vector3](#)s and writes the results into a specified range of a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (
    Vector3[] sourceArray,
    int sourceIndex,
    ref Matrix matrix,
    Vector3[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The source array.

sourceIndex

The index in the source array at which to start.

matrix

The transform [Matrix](#) to apply.

destinationArray

The existing destination array.

destinationIndex

The index in the destination array at which to start.

length

The number of [Vector3](#)s to transform.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3[], Int32, Quaternion, Vector3[], Int32, Int32)

Applies a specified [Quaternion](#) rotation to a specified range of an array of [Vector3s](#) and writes the results into a specified range of a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (
    Vector3[] sourceArray,
    int sourceIndex,
    ref Quaternion rotation,
    Vector3[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The source array.

sourceIndex

The index in the source array at which to start.

rotation

The [Quaternion](#) rotation to apply.

destinationArray

The existing destination array.

destinationIndex

The index in the destination array at which to start.

length

The number of [Vector3s](#) to transform.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3[], Matrix, Vector3[])

Transforms a source array of [Vector3](#)s by a specified [Matrix](#) and writes the results to an existing destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    Vector3[] sourceArray,  
    ref Matrix matrix,  
    Vector3[] destinationArray  
)
```

Parameters

sourceArray

The source array.

matrix

The transform [Matrix](#) to apply.

destinationArray

An existing destination array into which the transformed [Vector3](#)s are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Transform Method (Vector3[], Quaternion, Vector3[])

Transforms a source array of [Vector3s](#) by a specified [Quaternion](#) rotation and writes the results to an existing destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    Vector3[] sourceArray,  
    ref Quaternion rotation,  
    Vector3[] destinationArray  
)
```

Parameters

sourceArray

The source array.

rotation

The [Quaternion](#) rotation to apply.

destinationArray

An existing destination array into which the transformed [Vector3s](#) are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.TransformNormal Method

Transforms a vector normal by a matrix.

Overload List

Name	Description
Vector3.TransformNormal (Vector3, Matrix)	Transforms a 3D vector normal by a matrix.
Vector3.TransformNormal (Vector3, Matrix, Vector3)	Transforms a vector normal by a matrix.
Vector3.TransformNormal (Vector3[], Int32, Matrix, Vector3[], Int32, Int32)	Transforms a specified range in an array of 3D vector normals by a specified Matrix and writes the results to a specified range in a destination array.
Vector3.TransformNormal (Vector3[], Matrix, Vector3[])	Transforms an array of 3D vector normals by a specified Matrix .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.TransformNormal Method (Vector3, Matrix)

Transforms a 3D vector normal by a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 TransformNormal (  
    Vector3 normal,  
    Matrix matrix  
)
```

Parameters

normal

The source vector.

matrix

The transformation matrix.

Return Value

The transformed vector normal.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.TransformNormal Method (Vector3, Matrix, Vector3)

Transforms a vector normal by a matrix.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void TransformNormal (  
    ref Vector3 normal,  
    ref Matrix matrix,  
    out Vector3 result  
)
```

Parameters

normal

The source vector.

matrix

The transformation [Matrix](#).

result

[[OutAttribute](#)] The [Vector3](#) resulting from the transformation.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.TransformNormal Method (Vector3[], Int32, Matrix, Vector3[], Int32, Int32)

Transforms a specified range in an array of [3D vector](#) normals by a specified [Matrix](#) and writes the results to a specified range in a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void TransformNormal (
    Vector3[] sourceArray,
    int sourceIndex,
    ref Matrix matrix,
    Vector3[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The source array of [Vector3](#) normals.

sourceIndex

The starting index in the source array.

matrix

The transform [Matrix](#) to apply.

destinationArray

The destination [Vector3](#) array.

destinationIndex

The starting index in the destination array.

length

The number of vectors to transform.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.TransformNormal Method (Vector3[], Matrix, Vector3[])

Transforms an array of [3D vector](#) normals by a specified [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void TransformNormal (  
    Vector3[] sourceArray,  
    ref Matrix matrix,  
    Vector3[] destinationArray  
)
```

Parameters

sourceArray

The array of [Vector3](#) normals to transform.

matrix

The transform matrix to apply.

destinationArray

An existing [Vector3](#) array into which the results of the transforms are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector3 Structure](#)












[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3 Properties

Public Properties

	Name	Description
 S	Backward	Returns a unit Vector3 designating backward in a right-handed coordinate system (0, 0, 1).
 S	Down	Returns a unit Vector3 designating down (0, -1, 0).
 S	Forward	Returns a unit Vector3 designating forward in a right-handed coordinate system(0, 0, -1).
 S	Left	Returns a unit Vector3 designating left (-1, 0, 0).
 S	One	Returns a Vector3 with ones in all of its components.
 S	Right	Returns a unit Vector3 pointing to the right (1, 0, 0).
 S	UnitX	Returns the x unit Vector3 (1, 0, 0).
 S	UnitY	Returns the y unit Vector3 (0, 1, 0).
 S	UnitZ	Returns the z unit Vector3 (0, 0, 1).
 S	Up	Returns a unit vector designating up (0, 1, 0).
 S	Zero	Returns a Vector3 with all of its components set to zero.

See Also

Reference

[Vector3 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector3.Backward Property

Returns a unit [Vector3](#) designating backward in a right-handed coordinate system (0, 0, 1).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Backward { get; }
```

Property Value

A unit [Vector3](#) pointing backward.

Remarks

The XNA Framework uses a right-handed coordinate system, with the positive x-axis pointing to the right, the positive y-axis pointing up, and the positive z-axis pointing toward the observer.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Down Property

Returns a unit [Vector3](#) designating down (0, -1, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Down { get; }
```

Property Value

A unit [Vector3](#) pointing down.

Remarks

The XNA Framework uses a right-handed coordinate system, with the positive x-axis pointing to the right, the positive y-axis pointing up, and the positive z-axis pointing toward the observer.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Forward Property

Returns a unit [Vector3](#) designating forward in a right-handed coordinate system(0, 0, -1).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Forward { get; }
```

Property Value

A unit [Vector3](#) pointing forward.

Remarks

The XNA Framework uses a right-handed coordinate system, with the positive x-axis pointing to the right, the positive y-axis pointing up, and the positive z-axis pointing toward the observer.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Left Property

Returns a unit [Vector3](#) designating left (-1, 0, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Left { get; }
```

Property Value

A unit [Vector3](#) pointing left.

Remarks

The XNA Framework uses a right-handed coordinate system, with the positive x-axis pointing to the right, the positive y-axis pointing up, and the positive z-axis pointing toward the observer.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.One Property

Returns a [Vector3](#) with ones in all of its components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 One { get; }
```

Property Value

a [Vector3](#) with ones in all of its components.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Right Property

Returns a unit [Vector3](#) pointing to the right (1, 0, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Right { get; }
```

Property Value

A unit [Vector3](#) pointing to the right.

Remarks

The XNA Framework uses a right-handed coordinate system, with the positive x-axis pointing to the right, the positive y-axis pointing up, and the positive z-axis pointing toward the observer.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.UnitX Property

Returns the x unit [Vector3](#) (1, 0, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 UnitX { get; }
```

Property Value

Returns the x unit [Vector3](#) .

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Unity Property

Returns the y unit [Vector3](#) (0, 1, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Unity { get; }
```

Property Value

Returns the y unit [Vector3](#).

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.UnitZ Property

Returns the z unit [Vector3](#) (0, 0, 1).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 UnitZ { get; }
```

Property Value

Returns the z unit [Vector3](#).

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Up Property

Returns a unit vector designating up (0, 1, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Up { get; }
```

Property Value

A unit [Vector3](#) pointing up.

Remarks

The XNA Framework uses a right-handed coordinate system, with the positive x-axis pointing to the right, the positive y-axis pointing up, and the positive z-axis pointing toward the observer.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector3.Zero Property

Returns a [Vector3](#) with all of its components set to zero.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector3 Zero { get; }
```

Property Value

A [Vector3](#) with all of its components set to zero.

See Also

Reference

[Vector3 Structure](#)

[Vector3 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4 Structure

Defines a vector with four components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[TypeConverterAttribute("typeof(Microsoft.Xna.Framework.Design.Vector4Converter)")]  
[SerializableAttribute]  
public struct Vector4 : IEquatable<Vector4>
```

See Also

Reference

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Programming Guide


[Math Overview](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





Vector4 Members

The following tables list the members exposed by the Vector4 type.







Public Constructors

	Name	Description
	Vector4	Overloaded. Initializes a new instance of Vector4 .




















Public Fields














	Name	Description
	W	Gets or sets the w-component of the vector.
	X	Gets or sets the x-component of the vector.
	Y	Gets or sets the y-component of the vector.
	Z	Gets or sets the z-component of the vector.

Public Properties



	Name	Description
	One	Returns a Vector4 with all of its components set to one.
	UnitW	Returns the Vector4 (0, 0, 0, 1).
	UnitX	Returns the Vector4 (1, 0, 0, 0).
	UnitY	Returns the Vector4 (0, 1, 0, 0).
	UnitZ	Returns the Vector4 (0, 0, 1, 0).
	Zero	Returns a Vector4 with all of its components set to zero.

Public Methods

	Name	Description
	Add	Overloaded. Adds two vectors.
	Barycentric	Overloaded. Returns a Vector4 containing the 4D Cartesian coordinates of a point specified in barycentric (a real) coordinates relative to a 4D triangle.
	CatmullRom	Overloaded. Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Overloaded. Restricts a value to be within a specified range.
	Distance	Overloaded. Calculates the distance between two vectors.
	DistanceSquared	Overloaded. Calculates the distance between two vectors squared.
	Divide	Overloaded. Divide a vector by a scalar or another vector.
	Dot	Overloaded. Calculates the dot product of two vectors.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of this object.
	GetType	(Inherited from Object .)
	Hermite	Overloaded. Performs a Hermite spline interpolation.
	Length	Calculates the length of the vector.
	LengthSquared	Calculates the length of the vector squared.
	Lerp	Overloaded. Performs a linear interpolation between two vectors.
	Max	Overloaded. Returns a vector that contains the highest value from each matching pair of components.
	Min	Overloaded. Returns a vector that contains the lowest value from each matching pair of components.
	Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
	Negate	Overloaded. Returns a vector pointing in the opposite direction.

 Normalize	Overloaded. Creates a unit vector from the specified vector.
 op_Addition	Adds two vectors.
 op_Division	Overloaded. Divides a vector by a scalar or another vector.
 op_Equality	Tests vectors for equality.
 op_Inequality	Tests vectors for inequality.
 op_Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
 op_Subtraction	Subtracts a vector from a vector.
 op_UnaryNegation	Returns a vector pointing in the opposite direction.
 ReferenceEquals	(Inherited from Object .)
 SmoothStep	Overloaded. Interpolates between two values using a cubic equation.
 Subtract	Overloaded. Subtracts a vector from a vector.
 ToString	Retrieves a string representation of the current object.
 Transform	Overloaded. Transforms one or more vectors by a specified Matrix or Quaternion and returns the results in Vector4 form.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[Vector4 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4 Fields

Public Fields

	Name	Description
	W	Gets or sets the w-component of the vector.
	X	Gets or sets the x-component of the vector.
	Y	Gets or sets the y-component of the vector.
	Z	Gets or sets the z-component of the vector.

See Also

Reference

[Vector4 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.W Field

Gets or sets the w-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float W
```

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.X Field

Gets or sets the x-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float X
```

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Y Field

Gets or sets the y-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Y
```

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Z Field

Gets or sets the z-component of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Z
```

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4 Constructor

Initializes a new instance of [Vector4](#).

Overload List

Name	Description
Vector4 (Single)	Creates a new instance of Vector4 .
Vector4 (Single, Single, Single, Single)	Initializes a new instance of Vector4 .
Vector4 (Vector2, Single, Single)	Initializes a new instance of Vector4 .
Vector4 (Vector3, Single)	Initializes a new instance of Vector4 .

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4 Constructor (Single)

Creates a new instance of [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 (  
    float value  
)
```

Parameters

value

Value to initialize each component to.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4 Constructor (Single, Single, Single, Single)

Initializes a new instance of [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x-component of the vector.

y

Initial value for the y-component of the vector.

z

Initial value for the z-component of the vector.

w

Initial value for the w-component of the vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4 Constructor (Vector2, Single, Single)

Initializes a new instance of [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 (  
    Vector2 value,  
    float z,  
    float w  
)
```

Parameters

value

A vector containing the values to initialize x and y components with.

z

Initial value for the z-component of the vector.

w

Initial value for the w-component of the vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4 Constructor (Vector3, Single)

Initializes a new instance of [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 (  
    Vector3 value,  
    float w  
)
```

Parameters

value

A vector containing the values to initialize x, y, and z components with.

w

Initial value for the w-component of the vector.

See Also

Reference

[Vector4 Structure](#)































[Vector4 Members](#)




[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Vector4 Methods

Public Methods

	Name	Description
	Add	Overloaded. Adds two vectors.
	Barycentric	Overloaded. Returns a Vector4 containing the 4D Cartesian coordinates of a point specified in barycentric (a real) coordinates relative to a 4D triangle.
	CatmullRom	Overloaded. Performs a Catmull-Rom interpolation using the specified positions.
	Clamp	Overloaded. Restricts a value to be within a specified range.
	Distance	Overloaded. Calculates the distance between two vectors.
	DistanceSquared	Overloaded. Calculates the distance between two vectors squared.
	Divide	Overloaded. Divide a vector by a scalar or another vector.
	Dot	Overloaded. Calculates the dot product of two vectors.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code of this object.
	GetType	(Inherited from Object .)
	Hermite	Overloaded. Performs a Hermite spline interpolation.
	Length	Calculates the length of the vector.
	LengthSquared	Calculates the length of the vector squared.
	Lerp	Overloaded. Performs a linear interpolation between two vectors.
	Max	Overloaded. Returns a vector that contains the highest value from each matching pair of components.
	Min	Overloaded. Returns a vector that contains the lowest value from each matching pair of components.
	Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
	Negate	Overloaded. Returns a vector pointing in the opposite direction.
	Normalize	Overloaded. Creates a unit vector from the specified vector.
	op_Addition	Adds two vectors.
	op_Division	Overloaded. Divides a vector by a scalar or another vector.
	op_Equality	Tests vectors for equality.
	op_Inequality	Tests vectors for inequality.
	op_Multiply	Overloaded. Multiplies a vector by a scalar or another vector.
	op_Subtraction	Subtracts a vector from a vector.
	op_UnaryNegation	Returns a vector pointing in the opposite direction.
	ReferenceEquals	(Inherited from Object .)
	SmoothStep	Overloaded. Interpolates between two values using a cubic equation.
	Subtract	Overloaded. Subtracts a vector from a vector.

 ToString	Retrieves a string representation of the current object.
  Transform	Overloaded. Transforms one or more vectors by a specified Matrix or Quaternion and returns the results in Vector4 form.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Vector4 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Add Method

Adds two vectors.

Overload List

Name	Description
Vector4.Add (Vector4, Vector4)	Adds two vectors.
Vector4.Add (Vector4, Vector4, Vector4)	Adds two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Add Method (Vector4, Vector4)

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Add (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Sum of the two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Add Method (Vector4, Vector4, Vector4)

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Add (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] Sum of the source vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Barycentric Method

Returns a [Vector4](#) containing the 4D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 4D triangle.

Overload List

Name	Description
Vector4.Barycentric (Vector4, Vector4, Vector4, Single, Single)	Returns a Vector4 containing the 4D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 4D triangle.
Vector4.Barycentric (Vector4, Vector4, Vector4, Single, Single, Vector4)	Returns a Vector4 containing the 4D Cartesian coordinates of a point specified in Barycentric (areal) coordinates relative to a 4D triangle.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Barycentric Method (Vector4, Vector4, Vector4, Single, Single)

Returns a [Vector4](#) containing the 4D Cartesian coordinates of a point specified in barycentric (areal) coordinates relative to a 4D triangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Barycentric (
    Vector4 value1,
    Vector4 value2,
    Vector4 value3,
    float amount1,
    float amount2
)
```

Parameters

value1

A [Vector4](#) containing the 4D Cartesian coordinates of vertex 1 of the triangle.

value2

A [Vector4](#) containing the 4D Cartesian coordinates of vertex 2 of the triangle.

value3

A [Vector4](#) containing the 4D Cartesian coordinates of vertex 3 of the triangle.

amount1

Barycentric coordinate **b2**, which expresses the weighting factor toward vertex 2 (specified in *value2*).

amount2

Barycentric coordinate **b3**, which expresses the weighting factor toward vertex 3 (specified in *value3*).

Return Value

A new [Vector4](#) containing the 4D Cartesian coordinates of the specified point.

Remarks

About Barycentric Coordinates

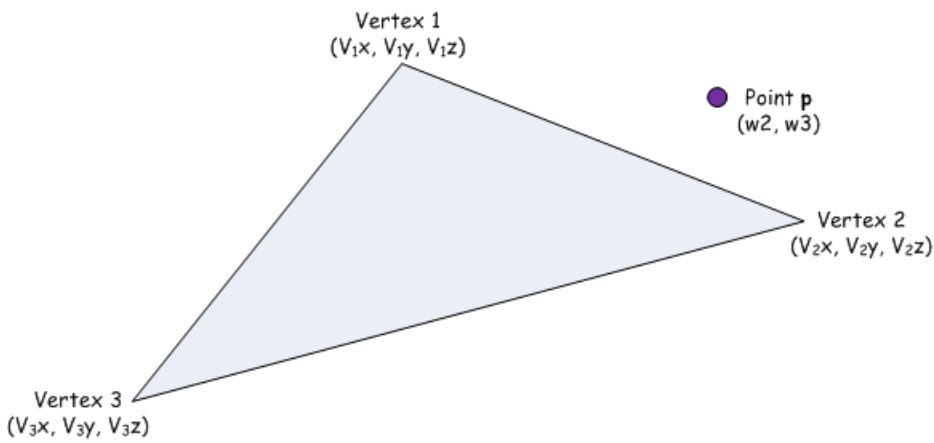
Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by 3 weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas.

$$\begin{aligned} P_x &= (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\ P_z &= (b_1 * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \\ P_w &= (b_1 * V_{1w}) + (b_2 * V_{2w}) + (b_3 * V_{3w}); \end{aligned}$$

Such triple weighting factors **b1**, **b2**, and **b3** are called *barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that **(k * b1)**, **(k * b2)**, and **(k * b3)** are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of barycentric coordinates is normalized so that **b1 + b2 + b3 = 1**, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** equals **(1 - b2 - b3)**.



What Vector4 Barycentric Does

The Vector4 [Barycentric](#) method takes three vectors specifying the Cartesian coordinates of the triangle vertices, **V1**, **V2**, and **V3**, and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

[Barycentric](#) then calculates the Cartesian coordinate of **P** as follows:

$$\begin{aligned}
 P_x &= ((1 - b_2 - b_3) * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\
 P_y &= ((1 - b_2 - b_3) * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\
 P_z &= ((1 - b_2 - b_3) * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \\
 P_w &= ((1 - b_2 - b_3) * V_{1w}) + (b_2 * V_{2w}) + (b_3 * V_{3w});
 \end{aligned}$$

Thus, to calculate the 3D Cartesian coordinates of **P**, you would pass the coordinates of the triangle vertices to [Barycentric](#) together with the appropriate normalized barycentric (areal) coordinates of **P**.

The following relationships may be useful.

- If $((amount1 \leq 0) \text{ and } (amount2 \geq 0) \text{ and } (1 - amount1 - amount2 \geq 0))$, then the point is inside the triangle defined by *value1*, *value2*, and *value3*.
- If $((amount1 == 0) \text{ and } (amount2 \geq 0) \text{ and } (1 - amount1 - amount2 \geq 0))$, then the point is on the line defined by *value1* and *value3*.
- If $((amount1 \geq 0) \text{ and } (amount2 == 0) \text{ and } (1 - amount1 - amount2 \geq 0))$, then the point is on the line defined by *value1* and *value2*.
- If $((amount1 \geq 0) \text{ and } (amount2 \geq 0) \text{ and } (1 - amount1 - amount2 == 0))$, then the point is on the line defined by *value2* and *value3*.

Barycentric coordinates are a form of general coordinates. In this context, using barycentric coordinates represents a change in coordinate systems. What holds true for Cartesian coordinates holds true for barycentric coordinates.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Barycentric Method (Vector4, Vector4, Vector4, Single, Single, Vector4)

Returns a [Vector4](#) containing the 4D Cartesian coordinates of a point specified in Barycentric (areal) coordinates relative to a 4D triangle.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Barycentric (
    ref Vector4 value1,
    ref Vector4 value2,
    ref Vector4 value3,
    float amount1,
    float amount2,
    out Vector4 result
)
```

Parameters

value1

A [Vector4](#) containing the 4D Cartesian coordinates of vertex 1 of the triangle.

value2

A [Vector4](#) containing the 4D Cartesian coordinates of vertex 2 of the triangle.

value3

A [Vector4](#) containing the 4D Cartesian coordinates of vertex 3 of the triangle.

amount1

Barycentric coordinate **b2**, which expresses the weighting factor toward vertex 2 (specified in *value2*).

amount2

Barycentric coordinate **b3**, which expresses the weighting factor toward vertex 3 (specified in *value3*).

result

[[OutAttribute](#)] The 4D Cartesian coordinates of the specified point are placed in this [Vector4](#) on exit.

Remarks

About Barycentric Coordinates

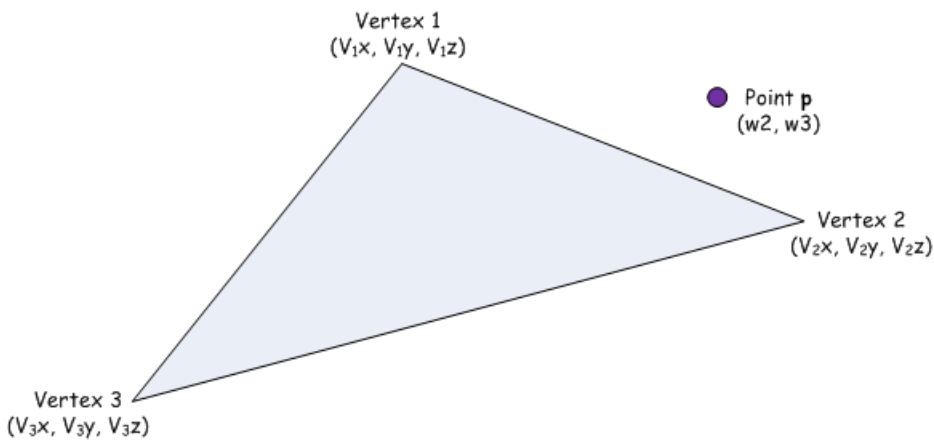
Given a triangle with vertices **V1**, **V2**, and **V3**, any point **P** on the plane of that triangle can be specified by three weighting factors **b1**, **b2**, and **b3**, each of which indicates how much relative influence the corresponding triangle vertex contributes to the location of the point, as specified in the following formulas.

$$\begin{aligned} P_x &= (b_1 * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}); \\ P_y &= (b_1 * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}); \\ P_z &= (b_1 * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}); \\ P_w &= (b_1 * V_{1w}) + (b_2 * V_{2w}) + (b_3 * V_{3w}); \end{aligned}$$

Such triple weighting factors **b1**, **b2**, and **b3** are called *barycentric coordinates*.

Barycentric coordinates express relative weights, meaning that **(k * b1)**, **(k * b2)**, and **(k * b3)** are also coordinates of the same point as **b1**, **b2**, and **b3** for any positive value of **k**.

If a set of barycentric coordinates is normalized so that **b1 + b2 + b3 = 1**, the resulting coordinates are unique for the point in question, and are known as *areal* coordinates. When normalized in this way, only two coordinates are needed, say **b2** and **b3**, since **b1** equals **(1 - b2 - b3)**.



What Vector4 Barycentric Does

The **Vector4 Barycentric** method takes three vectors specifying the Cartesian coordinates of the triangle vertices, **V1**, **V2**, and **V3**, and two areal coordinates **b2** and **b3** of some point **P** (**b2** is the *amount1* argument and **b3** is the *amount2* argument). The **b2** coordinate relates to vertex **V2**, and the **b3** coordinate relates to **V3**.

Barycentric then calculates the Cartesian coordinate of **P** as follows:

$$\begin{aligned}
 P_x &= ((1 - b_2 - b_3) * V_{1x}) + (b_2 * V_{2x}) + (b_3 * V_{3x}) ; \\
 P_y &= ((1 - b_2 - b_3) * V_{1y}) + (b_2 * V_{2y}) + (b_3 * V_{3y}) ; \\
 P_z &= ((1 - b_2 - b_3) * V_{1z}) + (b_2 * V_{2z}) + (b_3 * V_{3z}) ; \\
 P_w &= ((1 - b_2 - b_3) * V_{1w}) + (b_2 * V_{2w}) + (b_3 * V_{3w}) ;
 \end{aligned}$$

Thus, to calculate the 3D Cartesian coordinates of **P**, you would pass the coordinates of the triangle vertices to **Barycentric** together with the appropriate normalized barycentric (areal) coordinates of **P**.

The following relationships may be useful.

- If ((*amount1* <= 0) and (*amount2* >= 0) and (1 - *amount1* - *amount2* >= 0)), then the point is inside the triangle defined by *value1*, *value2*, and *value3*.
- If ((*amount1* == 0) and (*amount2* >= 0) and (1 - *amount1* - *amount2* >= 0)), then the point is on the line defined by *value1* and *value3*.
- If ((*amount1* >= 0) and (*amount2* == 0) and (1 - *amount1* - *amount2* >= 0)), then the point is on the line defined by *value1* and *value2*.
- If ((*amount1* >= 0) and (*amount2* >= 0) and (1 - *amount1* - *amount2* == 0)), then the point is on the line defined by *value2* and *value3*.

Barycentric coordinates are a form of general coordinates. In this context, using barycentric coordinates represents a change in coordinate systems. What holds true for Cartesian coordinates holds true for barycentric coordinates.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.CatmullRom Method

Performs a Catmull-Rom interpolation using the specified positions.

Overload List

Name	Description
Vector4.CatmullRom (Vector4, Vector4, Vector4, Vector4, Single)	Performs a Catmull-Rom interpolation using the specified positions.
Vector4.CatmullRom (Vector4, Vector4, Vector4, Vector4, Single, Vector4)	Performs a Catmull-Rom interpolation using the specified positions.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.CatmullRom Method (Vector4, Vector4, Vector4, Vector4, Single)

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 CatmullRom (  
    Vector4 value1,  
    Vector4 value2,  
    Vector4 value3,  
    Vector4 value4,  
    float amount  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

Return Value

A vector that is the result of the Catmull-Rom interpolation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.CatmullRom Method (Vector4, Vector4, Vector4, Vector4, Single, Vector4)

Performs a Catmull-Rom interpolation using the specified positions.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void CatmullRom (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    ref Vector4 value3,  
    ref Vector4 value4,  
    float amount,  
    out Vector4 result  
)
```

Parameters

value1

The first position in the interpolation.

value2

The second position in the interpolation.

value3

The third position in the interpolation.

value4

The fourth position in the interpolation.

amount

Weighting factor.

result

[[OutAttribute](#)] A vector that is the result of the Catmull-Rom interpolation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Clamp Method

Restricts a value to be within a specified range.

Overload List

Name	Description
Vector4.Clamp (Vector4, Vector4, Vector4)	Restricts a value to be within a specified range.
Vector4.Clamp (Vector4, Vector4, Vector4, Vector4)	Restricts a value to be within a specified range.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Clamp Method (Vector4, Vector4, Vector4)

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Clamp (  
    Vector4 value1,  
    Vector4 min,  
    Vector4 max  
)
```

Parameters

value1

The value to clamp.

min

The minimum value.

max

The maximum value.

Return Value

The clamped value.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Clamp Method (Vector4, Vector4, Vector4, Vector4)

Restricts a value to be within a specified range.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Clamp (  
    ref Vector4 value1,  
    ref Vector4 min,  
    ref Vector4 max,  
    out Vector4 result  
)
```

Parameters

value1

The value to clamp.

min

The minimum value.

max

The maximum value.

result

[[OutAttribute](#)] The clamped value.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Distance Method

Calculates the distance between two vectors.

Overload List

Name	Description
Vector4.Distance (Vector4, Vector4)	Calculates the distance between two vectors.
Vector4.Distance (Vector4, Vector4, Single)	Calculates the distance between two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Distance Method (Vector4, Vector4)

Calculates the distance between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Distance (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Distance between the source vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Distance Method (Vector4, Vector4, Single)

Calculates the distance between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Distance (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out float result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The distance between the vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.DistanceSquared Method

Calculates the distance between two vectors squared.

Overload List

Name	Description
Vector4.DistanceSquared (Vector4, Vector4)	Calculates the distance between two vectors squared.
Vector4.DistanceSquared (Vector4, Vector4, Single)	Calculates the distance between two vectors squared.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.DistanceSquared Method (Vector4, Vector4)

Calculates the distance between two vectors squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float DistanceSquared (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The distance between the source vectors squared.

Remarks

Distance is calculated with the following formula:

$$\sqrt{(a - b)^2}$$

Distance squared is the value before taking the square root. Distance squared can often be used in place of distance if relative comparisons are being made. For example, consider three points A, B, and C. To determine whether B or C is further from A, compare the distance between A and B to the distance between A and C. Calculating the two distances involves two square roots, which are computationally expensive. However, using distance squared provides the same information and avoids calculating two square roots.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.DistanceSquared Method (Vector4, Vector4, Single)

Calculates the distance between two vectors squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void DistanceSquared (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out float result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The distance between the two vectors squared.

Remarks

Distance is calculated with the formula:

$$\sqrt{(a - b)^2}$$

Distance squared is the value before taking the square root. Distance squared can often be used in place of distance if relative comparisons are being made. For example, consider three points A, B, and C. To determine whether B or C is further from A, compare the distance between A and B to the distance between A and C. Calculating the two distances involves two square roots, which are computationally expensive. However, using distance squared provides the same information and avoids calculating two square roots.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Divide Method

Divide a vector by a scalar or another vector.

Overload List

Name	Description
Vector4.Divide (Vector4, Single)	Divides a vector by a scalar value.
Vector4.Divide (Vector4, Single, Vector4)	Divides a vector by a scalar value.
Vector4.Divide (Vector4, Vector4)	Divides the components of a vector by the components of another vector.
Vector4.Divide (Vector4, Vector4, Vector4)	Divides the components of a vector by the components of another vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Divide Method (Vector4, Single)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Divide (  
    Vector4 value1,  
    float divider  
)
```

Parameters

value1

Source vector.

divider

The divisor.

Return Value

The source vector divided by *b*.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Divide Method (Vector4, Single, Vector4)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Vector4 value1,  
    float divider,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

divider

The divisor.

result

[[OutAttribute](#)] The result of the division.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Divide Method (Vector4, Vector4)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Divide (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Divisor vector.

Return Value

The result of dividing the vectors.

Remarks

Division of a vector by another vector is not mathematically defined. This method simply divides each component of *a* by the matching component of *b*.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Divide Method (Vector4, Vector4, Vector4)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Divide (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

The divisor.

result

[[OutAttribute](#)] The result of the division.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Dot Method

Calculates the dot product of two vectors.

Overload List

Name	Description
Vector4.Dot (Vector4, Vector4)	Calculates the dot product of two vectors.
Vector4.Dot (Vector4, Vector4, Single)	Calculates the dot product of two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Dot Method (Vector4, Vector4)

Calculates the dot product of two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Dot (  
    Vector4 vector1,  
    Vector4 vector2  
)
```

Parameters

vector1

Source vector.

vector2

Source vector.

Return Value

The dot product of the two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Dot Method (Vector4, Vector4, Single)

Calculates the dot product of two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Dot (  
    ref Vector4 vector1,  
    ref Vector4 vector2,  
    out float result  
)
```

Parameters

vector1

Source vector.

vector2

Source vector.

result

[[OutAttribute](#)] The dot product of the two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Vector4.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Vector4.Equals (Vector4)	Determines whether the specified Object is equal to the Vector4 .
Vector4.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Equals Method (Vector4)

Determines whether the specified [Object](#) is equal to the [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Vector4 other  
)
```

Parameters

other

The [Vector4](#) to compare with the current [Vector4](#).

Return Value

true if the specified [Vector4](#) is equal to the current [Vector4](#); **false** otherwise.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.GetHashCode Method

Gets the hash code of this object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code of the vector object.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Hermite Method

Performs a Hermite spline interpolation.

Overload List

Name	Description
Vector4.Hermite (Vector4, Vector4, Vector4, Vector4, Single)	Performs a Hermite spline interpolation.
Vector4.Hermite (Vector4, Vector4, Vector4, Vector4, Single, Vector4)	Performs a Hermite spline interpolation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Hermite Method (Vector4, Vector4, Vector4, Vector4, Single)

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Hermite (  
    Vector4 value1,  
    Vector4 tangent1,  
    Vector4 value2,  
    Vector4 tangent2,  
    float amount  
)
```

Parameters

value1

Source position vector.

tangent1

Source tangent vector.

value2

Source position vector.

tangent2

Source tangent vector.

amount

Weighting factor.

Return Value

The result of the Hermite spline interpolation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Hermite Method (Vector4, Vector4, Vector4, Vector4, Single, Vector4)

Performs a Hermite spline interpolation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Hermite (  
    ref Vector4 value1,  
    ref Vector4 tangent1,  
    ref Vector4 value2,  
    ref Vector4 tangent2,  
    float amount,  
    out Vector4 result  
)
```

Parameters

value1

Source position vector.

tangent1

Source tangent vector.

value2

Source position vector.

tangent2

Source tangent vector.

amount

Weighting factor.

result

[[OutAttribute](#)] The result of the Hermite spline interpolation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Length Method

Calculates the length of the vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Length ()
```

Return Value

The length of the vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.LengthSquared Method

Calculates the length of the vector squared.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float LengthSquared ()
```

Return Value

The length of the vector squared.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Lerp Method

Performs a linear interpolation between two vectors.

Overload List

Name	Description
Vector4.Lerp (Vector4, Vector4, Single)	Performs a linear interpolation between two vectors.
Vector4.Lerp (Vector4, Vector4, Single, Vector4)	Performs a linear interpolation between two vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Lerp Method (Vector4, Vector4, Single)

Performs a linear interpolation between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Lerp (  
    Vector4 value1,  
    Vector4 value2,  
    float amount  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Value between 0 and 1 indicating the weight of *value2*.

Return Value

The linear interpolation of the two vectors.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned; a value of 1 will cause *value2* to be returned.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Lerp Method (Vector4, Vector4, Single, Vector4)

Performs a linear interpolation between two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Lerp (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    float amount,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Value between 0 and 1 indicating the weight of *value2*.

result

[[OutAttribute](#)] The result of the interpolation.

Remarks

This method performs the linear interpolation based on the following formula.

$$\text{value1} + (\text{value2} - \text{value1}) * \text{amount}$$

Passing *amount* a value of 0 will cause *value1* to be returned; a value of 1 will cause *value2* to be returned.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Max Method

Returns a vector that contains the highest value from each matching pair of components.

Overload List

Name	Description
Vector4.Max (Vector4, Vector4)	Returns a vector that contains the highest value from each matching pair of components.
Vector4.Max (Vector4, Vector4, Vector4)	Returns a vector that contains the highest value from each matching pair of components.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Max Method (Vector4, Vector4)

Returns a vector that contains the highest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Max (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The maximized vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Max Method (Vector4, Vector4, Vector4)

Returns a vector that contains the highest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Max (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The maximized vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Min Method

Returns a vector that contains the lowest value from each matching pair of components.

Overload List

Name	Description
Vector4.Min (Vector4, Vector4)	Returns a vector that contains the lowest value from each matching pair of component s.
Vector4.Min (Vector4, Vector4, Vector4)	Returns a vector that contains the lowest value from each matching pair of component s.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Min Method (Vector4, Vector4)

Returns a vector that contains the lowest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Min (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

The minimized vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Min Method (Vector4, Vector4, Vector4)

Returns a vector that contains the lowest value from each matching pair of components.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Min (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The minimized vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Multiply Method

Multiplies a vector by a scalar or another vector.

Overload List

Name	Description
Vector4.Multiply (Vector4, Single)	Multiplies a vector by a scalar.
Vector4.Multiply (Vector4, Single, Vector4)	Multiplies a vector by a scalar value.
Vector4.Multiply (Vector4, Vector4)	Multiplies the components of two vectors by each other.
Vector4.Multiply (Vector4, Vector4, Vector4)	Multiplies the components of two vectors by each other.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Multiply Method (Vector4, Single)

Multiplies a vector by a scalar.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Multiply (  
    Vector4 value1,  
    float scaleFactor  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Multiply Method (Vector4, Single, Vector4)

Multiplies a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Vector4 value1,  
    float scaleFactor,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Multiply Method (Vector4, Vector4)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Multiply (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the multiplication.

Remarks

Multiplication performed by this method is not vector multiplication (dot product and cross product) but multiplication of the corresponding components of each vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Multiply Method (Vector4, Vector4, Vector4)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Multiply (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The result of the multiplication.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Negate Method

Returns a vector pointing in the opposite direction.

Overload List

Name	Description
Vector4.Negate (Vector4)	Returns a vector pointing in the opposite direction.
Vector4.Negate (Vector4, Vector4)	Returns a vector pointing in the opposite direction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Negate Method (Vector4)

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Negate (  
    Vector4 value  
)
```

Parameters

value

Source vector.

Return Value

Vector pointing in the opposite direction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Negate Method (Vector4, Vector4)

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Negate (  
    ref Vector4 value,  
    out Vector4 result  
)
```

Parameters

value

Source vector.

result

[[OutAttribute](#)] Vector pointing in the opposite direction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Normalize Method

Creates a unit vector from the specified vector.

Overload List

Name	Description
Vector4.Normalize ()	Turns the current vector into a unit vector.
Vector4.Normalize (Vector4)	Creates a unit vector from the specified vector.
Vector4.Normalize (Vector4, Vector4)	Returns a normalized version of the specified vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Normalize Method ()

Turns the current vector into a unit vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Normalize ()
```

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Normalize Method (Vector4)

Creates a unit vector from the specified vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Normalize (  
    Vector4 vector  
)
```

Parameters

vector

The source [Vector4](#).

Return Value

The created unit vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Normalize Method (Vector4, Vector4)

Returns a normalized version of the specified vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Normalize (  
    ref Vector4 vector,  
    out Vector4 result  
)
```

Parameters

vector

Source vector.

result

[[OutAttribute](#)] The normalized vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Addition Method

Adds two vectors.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Addition (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Sum of the vectors.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Division Method

Divides a vector by a scalar or another vector.

Overload List

Name	Description
Vector4.op_Division (Vector4, Single)	Divides a vector by a scalar value.
Vector4.op_Division (Vector4, Vector4)	Divides the components of a vector by the components of another vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.op_Division Method (Vector4, Single)

Divides a vector by a scalar value.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Division (  
    Vector4 value1,  
    float divider  
)
```

Parameters

value1

Source vector.

divider

The divisor.

Return Value

The source vector divided by *div*.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Division Method (Vector4, Vector4)

Divides the components of a vector by the components of another vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Division (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Divisor vector.

Return Value

The result of dividing the vectors.

Remarks

Division of a vector by another vector is not mathematically defined. This method simply divides each component of *a* by the matching component of *b*.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Equality Method

Tests vectors for equality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

true if the vectors are equal; **false** otherwise.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Inequality Method

Tests vectors for inequality.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Vector to compare.

value2

Vector to compare.

Return Value

true if the vectors are not equal; **false** otherwise.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Multiply Method

Multiplies a vector by a scalar or another vector.

Overload List

Name	Description
Vector4.op_Multiply (Single, Vector4)	Multiplies a vector by a scalar.
Vector4.op_Multiply (Vector4, Single)	Multiplies a vector by a scalar.
Vector4.op_Multiply (Vector4, Vector4)	Multiplies the components of two vectors by each other.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.op_Multiply Method (Single, Vector4)

Multiplies a vector by a scalar.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Multiply (  
    float scaleFactor,  
    Vector4 value1  
)
```

Parameters

scaleFactor

Scalar value.

value1

Source vector.

Return Value

Result of the multiplication.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Multiply Method (Vector4, Single)

Multiplies a vector by a scalar.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Multiply (  
    Vector4 value1,  
    float scaleFactor  
)
```

Parameters

value1

Source vector.

scaleFactor

Scalar value.

Return Value

Result of the multiplication.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Multiply Method (Vector4, Vector4)

Multiplies the components of two vectors by each other.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Multiply (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the multiplication.

Remarks

Multiplication performed by this method is not vector multiplication (dot product and cross product) but multiplication of the corresponding components of each vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_Subtraction Method

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_Subtraction (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the subtraction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.op_UnaryNegation Method

Returns a vector pointing in the opposite direction.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 op_UnaryNegation (  
    Vector4 value  
)
```

Parameters

value

Source vector.

Return Value

Vector pointing in the opposite direction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.SmoothStep Method

Interpolates between two values using a cubic equation.

Overload List

Name	Description
Vector4.SmoothStep (Vector4, Vector4, Single)	Interpolates between two values using a cubic equation.
Vector4.SmoothStep (Vector4, Vector4, Single, Vector4)	Interpolates between two values using a cubic equation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.SmoothStep Method (Vector4, Vector4, Single)

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 SmoothStep (  
    Vector4 value1,  
    Vector4 value2,  
    float amount  
)
```

Parameters

value1

Source value.

value2

Source value.

amount

Weighting value.

Return Value

Interpolated value.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.SmoothStep Method (Vector4, Vector4, Single, Vector4)

Interpolates between two values using a cubic equation.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void SmoothStep (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    float amount,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

amount

Weighting factor.

result

[[OutAttribute](#)] The interpolated value.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Subtract Method

Subtracts a vector from a vector.

Overload List

Name	Description
Vector4.Subtract (Vector4, Vector4)	Subtracts a vector from a vector.
Vector4.Subtract (Vector4, Vector4, Vector4)	Subtracts a vector from a vector.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Subtract Method (Vector4, Vector4)

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Subtract (  
    Vector4 value1,  
    Vector4 value2  
)
```

Parameters

value1

Source vector.

value2

Source vector.

Return Value

Result of the subtraction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Subtract Method (Vector4, Vector4, Vector4)

Subtracts a vector from a vector.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Subtract (  
    ref Vector4 value1,  
    ref Vector4 value2,  
    out Vector4 result  
)
```

Parameters

value1

Source vector.

value2

Source vector.

result

[[OutAttribute](#)] The result of the subtraction.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.ToString Method

Retrieves a string representation of the current object.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method

Transforms one or more vectors by a specified [Matrix](#) or [Quaternion](#) and returns the results in [Vector4](#) form.

Overload List

Name	Description
Vector4.Transform (Vector2, Matrix)	Transforms a Vector2 by the given Matrix .
Vector4.Transform (Vector2, Matrix, Vector4)	Transforms a Vector2 by the given Matrix .
Vector4.Transform (Vector2, Quaternion)	Transforms a Vector2 by a specified Quaternion into a Vector4 .
Vector4.Transform (Vector2, Quaternion, Vector4)	Transforms a Vector2 by a specified Quaternion into a Vector4 .
Vector4.Transform (Vector3, Matrix)	Transforms a Vector3 by the given Matrix .
Vector4.Transform (Vector3, Matrix, Vector4)	Transforms a Vector3 by the given Matrix .
Vector4.Transform (Vector3, Quaternion)	Transforms a Vector3 by a specified Quaternion into a Vector4 .
Vector4.Transform (Vector3, Quaternion, Vector4)	Transforms a Vector3 by a specified Quaternion into a Vector4 .
Vector4.Transform (Vector4, Matrix)	Transforms a Vector4 by the specified Matrix .
Vector4.Transform (Vector4, Matrix, Vector4)	Transforms a Vector4 by the given Matrix .
Vector4.Transform (Vector4, Quaternion)	Transforms a Vector4 by a specified Quaternion .
Vector4.Transform (Vector4, Quaternion, Vector4)	Transforms a Vector4 by a specified Quaternion .
Vector4.Transform (Vector4[], Int32, Matrix, Vector4[], Int32, Int32)	Transforms a specified range in an array of Vector4 s by a specified Matrix into a specified range in a destination array.
Vector4.Transform (Vector4[], Int32, Quaternion, Vector4[], Int32, Int32)	Transforms a specified range in an array of Vector4 s by a specified Quaternion into a specified range in a destination array.
Vector4.Transform (Vector4[], Matrix, Vector4[])	Transforms an array of Vector4 s by a specified Matrix .
Vector4.Transform (Vector4[], Quaternion, Vector4[])	Transforms an array of Vector4 s by a specified Quaternion .

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.Transform Method (Vector2, Matrix)

Transforms a [Vector2](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Transform (  
    Vector2 position,  
    Matrix matrix  
)
```

Parameters

position

The source [Vector2](#).

matrix

The transformation [Matrix](#).

Return Value

The transformed [Vector4](#).

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector2, Matrix, Vector4)

Transforms a [Vector2](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector2 position,  
    ref Matrix matrix,  
    out Vector4 result  
)
```

Parameters

position

The source [Vector2](#).

matrix

The transformation [Matrix](#).

result

[[OutAttribute](#)] The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector2, Quaternion)

Transforms a [Vector2](#) by a specified [Quaternion](#) into a [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Transform (  
    Vector2 value,  
    Quaternion rotation  
)
```

Parameters

value

The [Vector2](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

Return Value

Returns the [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector2, Quaternion, Vector4)

Transforms a [Vector2](#) by a specified [Quaternion](#) into a [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector2 value,  
    ref Quaternion rotation,  
    out Vector4 result  
)
```

Parameters

value

The [Vector2](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

result

[[OutAttribute](#)] The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector3, Matrix)

Transforms a [Vector3](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Transform (  
    Vector3 position,  
    Matrix matrix  
)
```

Parameters

position

The source [Vector3](#).

matrix

The transformation [Matrix](#).

Return Value

The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector3, Matrix, Vector4)

Transforms a [Vector3](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector3 position,  
    ref Matrix matrix,  
    out Vector4 result  
)
```

Parameters

position

The source [Vector3](#).

matrix

The transformation [Matrix](#).

result

[[OutAttribute](#)] The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector3, Quaternion)

Transforms a [Vector3](#) by a specified [Quaternion](#) into a [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Transform (  
    Vector3 value,  
    Quaternion rotation  
)
```

Parameters

value

The [Vector3](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

Return Value

The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector3, Quaternion, Vector4)

Transforms a [Vector3](#) by a specified [Quaternion](#) into a [Vector4](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector3 value,  
    ref Quaternion rotation,  
    out Vector4 result  
)
```

Parameters

value

The [Vector3](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

result

[[OutAttribute](#)] The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4, Matrix)

Transforms a [Vector4](#) by the specified [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Transform (  
    Vector4 vector,  
    Matrix matrix  
)
```

Parameters

vector

The source [Vector4](#).

matrix

The transformation [Matrix](#).

Return Value

The transformed [Vector4](#).

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4, Matrix, Vector4)

Transforms a [Vector4](#) by the given [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector4 vector,  
    ref Matrix matrix,  
    out Vector4 result  
)
```

Parameters

vector

The source [Vector4](#).

matrix

The transformation [Matrix](#).

result

[[OutAttribute](#)] The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4, Quaternion)

Transforms a [Vector4](#) by a specified [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Transform (  
    Vector4 value,  
    Quaternion rotation  
)
```

Parameters

value

The [Vector4](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

Return Value

The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4, Quaternion, Vector4)

Transforms a [Vector4](#) by a specified [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    ref Vector4 value,  
    ref Quaternion rotation,  
    out Vector4 result  
)
```

Parameters

value

The [Vector4](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

result

[[OutAttribute](#)] The [Vector4](#) resulting from the transformation.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4[], Int32, Matrix, Vector4[], Int32, Int32)

Transforms a specified range in an array of [Vector4s](#) by a specified [Matrix](#) into a specified range in a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (
    Vector4[] sourceArray,
    int sourceIndex,
    ref Matrix matrix,
    Vector4[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The array of [Vector4s](#) containing the range to transform.

sourceIndex

The index in the source array of the first [Vector4](#) to transform.

matrix

The transform [Matrix](#) to apply.

destinationArray

The existing destination array of [Vector4s](#) into which to write the results.

destinationIndex

The index in the destination array of the first result [Vector4](#) to write.

length

The number of [Vector4s](#) to transform.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4[], Int32, Quaternion, Vector4[], Int32, Int32)

Transforms a specified range in an array of [Vector4s](#) by a specified [Quaternion](#) into a specified range in a destination array.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (
    Vector4[] sourceArray,
    int sourceIndex,
    ref Quaternion rotation,
    Vector4[] destinationArray,
    int destinationIndex,
    int length
)
```

Parameters

sourceArray

The array of [Vector4s](#) containing the range to transform.

sourceIndex

The index in the source array of the first [Vector4](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

destinationArray

The existing destination array of [Vector4s](#) into which to write the results.

destinationIndex

The index in the destination array of the first result [Vector4](#) to write.

length

The number of [Vector4s](#) to transform.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or the combination of <i>sourceIndex</i> and <i>length</i> was greater than <i>sourceArray.Length</i> .
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4[], Matrix, Vector4[])

Transforms an array of [Vector4s](#) by a specified [Matrix](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    Vector4[] sourceArray,  
    ref Matrix matrix,  
    Vector4[] destinationArray  
)
```

Parameters

sourceArray

The array of [Vector4s](#) to transform.

matrix

The transform [Matrix](#) to apply.

destinationArray

The existing destination array into which the transformed [Vector4s](#) are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Transform Method (Vector4[], Quaternion, Vector4[])

Transforms an array of [Vector4s](#) by a specified [Quaternion](#).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Transform (  
    Vector4[] sourceArray,  
    ref Quaternion rotation,  
    Vector4[] destinationArray  
)
```

Parameters

sourceArray

The array of [Vector4s](#) to transform.

rotation

The [Quaternion](#) rotation to apply.

destinationArray

The existing destination array into which the transformed [Vector4s](#) are written.

Exceptions

Exception type	Condition
ArgumentException	<i>destinationArray</i> is too small to contain the result or.
ArgumentNullException	<i>sourceArray</i> or <i>destinationArray</i> is null .

See Also

Reference

[Vector4 Structure](#)







[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4 Properties

Public Properties

	Name	Description
 S	One	Returns a Vector4 with all of its components set to one.
 S	UnitW	Returns the Vector4 (0, 0, 0, 1).
 S	UnitX	Returns the Vector4 (1, 0, 0, 0).
 S	UnitY	Returns the Vector4 (0, 1, 0, 0).
 S	UnitZ	Returns the Vector4 (0, 0, 1, 0).
 S	Zero	Returns a Vector4 with all of its components set to zero.

See Also

Reference

[Vector4 Structure](#)

[Microsoft.Xna.Framework Namespace](#)

Vector4.One Property

Returns a [Vector4](#) with all of its components set to one.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 One { get; }
```

Property Value

A [Vector4](#) with all of its components set to one.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.UnitW Property

Returns the [Vector4](#) (0, 0, 0, 1).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 UnitW { get; }
```

Property Value

The [Vector4](#) (0, 0, 0, 1).

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.UnitX Property

Returns the [Vector4](#) (1, 0, 0, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 UnitX { get; }
```

Property Value

The [Vector4](#) (1, 0, 0, 0).

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Unity Property

Returns the [Vector4](#) (0, 1, 0, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Unity { get; }
```

Property Value

The [Vector4](#) (0, 1, 0, 0).

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.UnitZ Property

Returns the [Vector4](#) (0, 0, 1, 0).

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 UnitZ { get; }
```

Property Value

The [Vector4](#) (0, 0, 1, 0).

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)

[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Vector4.Zero Property

Returns a [Vector4](#) with all of its components set to zero.

Namespace: Microsoft.Xna.Framework

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Vector4 Zero { get; }
```

Property Value

A [Vector4](#) with all of its components set to zero.

See Also

Reference

[Vector4 Structure](#)

[Vector4 Members](#)











[Microsoft.Xna.Framework Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Microsoft.Xna.Framework.Audio Namespace

Contains low-level application programming interface (API) methods that can load and manipulate XACT-created project and content files to play audio.



Classes

Name	Description
 AudioEmitter	Represents a 3D audio emitter.
 AudioEngine	Represents the audio engine. Applications use the methods of the audio engine to instantiate and manipulate core audio objects.
 AudioListener	Represents a 3D audio listener.
 Cue	Defines methods for managing the playback of sounds.
 InstancePlayLimitException	The exception that is thrown when there is an attempt to play more than 16 SoundEffectInstance sounds concurrently.
 NoAudioHardwareException	The exception that is thrown when no audio hardware is present, or when audio hardware is installed, but the device drivers for the audio hardware are not present or enabled.
 SoundBank	Represents a sound bank, which is a collection of cues.
 SoundEffect	Provides a loaded sound resource.
 SoundEffectInstance	Provides a single playing, paused, or stopped instance of a SoundEffect sound.
 WaveBank	Represents a wave bank, which is a collection of wave files.

Structures

Name	Description
 AudioCategory	Represents a particular category of sounds.
 RendererDetail	Represents an audio renderer, which is a device that can render audio to a user.

Enumerations

Name	Description
 AudioStopOptions	Controls how Cue objects should stop when Stop is called.
 SoundState	Current state (playing, paused, or stopped) of a SoundEffectInstance .

See Also

Tasks

- [How To: Play a Sound](#)
- [How To: Play a Song](#)
- [How To: Stream a Sound Using XACT](#)
- [How To: Add a Sound File to Your Game Using XACT](#)
- [How To: Play a Sound Using XACT](#)
- [How To: Specify Background Music Using XACT](#)
- [How To: Stop or Pause a Sound Using XACT](#)
- [How To: Change Sound Volume Levels Using XACT](#)
- [How To: Apply Basic 3D Positional Effects to a Cue](#)
- [How To: Apply Attenuation and Doppler 3D Audio Effects](#)

Concepts

- [Audio Overview](#)
- [Audio Content Catalog at XNA Creators Club Online](#)

AudioCategory Structure

Represents a particular category of sounds.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct AudioCategory
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[AudioCategory Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista












AudioCategory Members

The following tables list the members exposed by the AudioCategory type.



Public Properties

	Name	Description
	Name	Specifies the friendly name of this category.

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of AudioCategory are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Sop_Equality	Determines whether the specified AudioCategory instances are equal.
	Sop_Inequality	Determines whether the specified AudioCategory instances are not equal.
	Pause	Pauses all sounds associated with this category.
	ReferenceEquals	(Inherited from Object .)
	Resume	Resumes all paused sounds associated with this category.
	SetVolume	Sets the volume of all sounds associated with this category.
	Stop	Stops all sounds associated with this category.
	ToString	Returns a String representation of this AudioCategory .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Concepts

[Audio Overview](#)












Reference

[AudioCategory Structure](#)



[Microsoft.Xna.Framework.Audio Namespace](#)

AudioCategory Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of AudioCategory are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Sop_Equality	Determines whether the specified AudioCategory instances are equal.
	Sop_Inequality	Determines whether the specified AudioCategory instances are not equal.
	Pause	Pauses all sounds associated with this category.
	ReferenceEquals	(Inherited from Object .)
	Resume	Resumes all paused sounds associated with this category.
	SetVolume	Sets the volume of all sounds associated with this category.
	Stop	Stops all sounds associated with this category.
	ToString	Returns a String representation of this AudioCategory .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AudioCategory Structure](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioCategory.Equals Method

Determines whether two instances of [AudioCategory](#) are equal.

Overload List

Name	Description
AudioCategory.Equals (AudioCategory)	Determines whether the specified AudioCategory is equal to this AudioCategory .
AudioCategory.Equals (Object)	Determines whether the specified Object is equal to this AudioCategory .
AudioCategory.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioCategory.Equals Method (AudioCategory)

Determines whether the specified [AudioCategory](#) is equal to this [AudioCategory](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    AudioCategory other  
)
```

Parameters

other

[AudioCategory](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [AudioCategory](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.op_Equality Method

Determines whether the specified [AudioCategory](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    AudioCategory value1,  
    AudioCategory value2  
)
```

Parameters

value1

Object to the left of the equality operator.

value2

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.op_Inequality Method

Determines whether the specified [AudioCategory](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    AudioCategory value1,  
    AudioCategory value2  
)
```

Parameters

value1

Object to the left of the inequality operator.

value2

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.Pause Method

Pauses all sounds associated with this category.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Pause ()
```

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.Resume Method

Resumes all paused sounds associated with this category.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Resume ()
```

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.SetVolume Method

Sets the volume of all sounds associated with this category.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVolume (  
    float volume  
)
```

Parameters

volume

Volume amplitude multiplier. *volume* is normally between 0.0 (silence) and 1.0 (full volume), but can range from 0.0f to `float.MaxValue`.

Volume levels map to decibels (dB) as shown in the following table.

Volume	Description
0.0f	-96 dB (silence)
1.0f	+0 dB (full volume as authored)
2.0f	+6 dB (6 dB greater than authored)

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.Stop Method

Stops all sounds associated with this category.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Stop (  
    AudioStopOptions options  
)
```

Parameters

options

Enumerated value specifying how the sounds should be stopped.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory.ToString Method

Returns a [String](#) representation of this [AudioCategory](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[AudioCategory Structure](#)


[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioCategory Properties

Public Properties

	Name	Description
	Name	Specifies the friendly name of this category.

See Also

Reference

[AudioCategory Structure](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioCategory.Name Property

Specifies the friendly name of this category.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

Friendly name of this category.

See Also

Reference

[AudioCategory Structure](#)

[AudioCategory Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEmitter Class

Represents a 3D audio emitter. This object, used in combination with an [AudioListener](#), can simulate 3D audio effects for a given [Cue](#) or [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class AudioEmitter
```

See Also

Tasks

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)

Reference

[AudioEmitter Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






AudioEmitter Members

The following tables list the members exposed by the AudioEmitter type.






Public Constructors

Name	Description
 AudioEmitter	Initializes a new instance of this class.



Public Properties

Name	Description
 DopplerScale	Gets or sets a scalar applied to the level of Doppler effect calculated between this and any AudioListener .
 Forward	Gets or sets the forward orientation vector for this emitter.
 Position	Gets or sets the position of this emitter.
 Up	Gets or sets the upward orientation vector for this emitter.
 Velocity	Gets or sets the velocity vector of this emitter.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AudioEmitter Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEmitter Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AudioEmitter ()
```

See Also

Reference

[AudioEmitter Class](#)






[AudioEmitter Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioEmitter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[AudioEmitter Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEmitter Properties

Public Properties

	Name	Description
	DopplerScale	Gets or sets a scalar applied to the level of Doppler effect calculated between this and any AudioListener .
	Forward	Gets or sets the forward orientation vector for this emitter.
	Position	Gets or sets the position of this emitter.
	Up	Gets or sets the upward orientation vector for this emitter.
	Velocity	Gets or sets the velocity vector of this emitter.

See Also

Reference

[AudioEmitter Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEmitter.DopplerScale Property

Gets or sets a scalar applied to the level of Doppler effect calculated between this and any [AudioListener](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float DopplerScale { get; set; }
```

Property Value

The Doppler scale value.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The doppler scale of an audio emitter must be greater than or equal to zero.

Remarks

By default, this value is 1.0.

This value determines how much to modify the calculated Doppler effect between this object and a [AudioListener](#). Values below 1.0 scale down the Doppler effect to make it less apparent. Values above 1.0 exaggerate the Doppler effect. A value of 1.0 leaves the effect unmodified.

Note that this value modifies only the calculated Doppler between this object and a [AudioListener](#). The calculated Doppler is a product of the relationship between [AudioEmitter.Velocity](#) and [AudioListener.Velocity](#). If the calculation yields a result of no Doppler effect, this value has no effect.

See Also

Reference

[AudioEmitter Class](#)

[AudioEmitter Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioEmitter.Forward Property

Gets or sets the forward orientation vector for this emitter.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Forward { get; set; }
```

Property Value

The forward orientation vector.

Remarks

By default, this value is [Vector3.Forward](#). In most cases, it is not necessary to modify this value.

Doppler values between an [AudioEmitter](#) and an [AudioListener](#) are calculated by the relationship between [AudioListener.Velocity](#) and [AudioEmitter.Velocity](#) with respect to the axes defined by the [Forward](#) and [Up](#) vectors of each.

The values of the [Forward](#) and [Up](#) vectors must be orthonormal (at right angles to one another). Behavior is undefined if these vectors are not orthonormal.

See Also

Reference

[AudioEmitter Class](#)

[AudioEmitter Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioEmitter.Position Property

Gets or sets the position of this emitter.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position { get; set; }
```

Property Value

The position vector.

Remarks

By default, the value of this property is [Vector3.Zero](#).

The relative positions of an [AudioEmitter](#) and an [AudioListener](#) are used to determine speaker positioning of a sound.

See Also

Tasks

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)

Reference

[AudioEmitter Class](#)

[AudioEmitter Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioEmitter.Up Property

Gets or sets the upward orientation vector for this emitter.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Up { get; set; }
```

Property Value

The upward orientation vector.

Remarks

By default, this value is [Vector3.Up](#). In most cases, it is not necessary to modify this value.

Doppler values between an [AudioEmitter](#) and an [AudioListener](#) are calculated by the relationship between [AudioListener.Velocity](#) and [AudioEmitter.Velocity](#) with respect to the axes defined by the [Forward](#) and [Up](#) vectors of each.

The values of the [Forward](#) and [Up](#) vectors must be orthonormal (at right angles to one another). Behavior is undefined if these vectors are not orthonormal.

See Also

Reference

[AudioEmitter Class](#)

[AudioEmitter Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioEmitter.Velocity Property

Gets or sets the velocity vector of this emitter.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Velocity { get; set; }
```

Property Value

The velocity vector.

Remarks

By default, the value of this property is [Vector3.Zero](#).

The Doppler effect value applied to a [Cue](#) is based on the relative [Velocity](#) values of the [AudioEmitter](#) and [AudioListener](#), scaled by the [DopplerScale](#) value.

The [Velocity](#) property is used only to calculate Doppler values. It is not applied to the [Position](#) vector or otherwise used to set game state. You must set [Velocity](#) and [Position](#) each frame to maintain accurate 3D audio values in your game.

See Also

Reference

[AudioEmitter Class](#)

[AudioEmitter Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioEngine Class

Represents the audio engine. Applications use the methods of the audio engine to instantiate and manipulate core audio objects.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class AudioEngine : IDisposable
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[AudioEngine Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


AudioEngine Members

The following tables list the members exposed by the AudioEngine type.



Public Constructors

Name	Description
 AudioEngine	Overloaded. Initializes a new instance of this class.











Public Fields

Name	Description
 ContentVersion	Specifies the current content version.



Public Properties

Name	Description
 IsDisposed	Gets a value that indicates whether the object is disposed.
 RendererDetails	Gets a collection of audio renderers.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetCategory	Gets an audio category.
 GetGlobalVariable	Gets the value of a global variable.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetGlobalVariable	Sets the value of a global variable.
 ToString	(Inherited from Object .)
 Update	Performs periodic work required by the audio engine.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Concepts

[Audio Overview](#)


Reference

[AudioEngine Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine Fields

Public Fields

	Name	Description
	ContentVersion	Specifies the current content version.

See Also

Reference

[AudioEngine Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine.ContentVersion Field

Specifies the current content version.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int ContentVersion
```

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

AudioEngine Constructor

Initializes a new instance of this class.

Overload List

Name	Description
AudioEngine (String)	Initializes a new instance of this class, using a path to an XACT global settings file.
AudioEngine (String, TimeSpan, String)	Initializes a new instance of this class, using a settings file, a specific audio renderer, and a specific speaker configuration.

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine Constructor (String)

Initializes a new instance of this class, using a path to an XACT global settings file.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AudioEngine (  
    string settingsFile  
)
```

Parameters

settingsFile

Path to a global settings file.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>settingsFile</i> parameter is null .
ArgumentException	XACT could not load the data provided. Make sure you are using the correct version of the XACT tool.
InvalidOperationException	The engine could not be created.

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine Constructor (String, TimeSpan, String)

Initializes a new instance of this class, using a settings file, a specific audio renderer, and a specific speaker configuration.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AudioEngine (
    string settingsFile,
    TimeSpan lookAheadTime,
    string rendererId
)
```

Parameters

settingsFile

Path to a global settings file.

lookAheadTime

Interactive audio and branch event look-ahead time, in milliseconds.

rendererId

A string that specifies the audio renderer to use.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>settingsFile</i> parameter is null .
ArgumentException	XACT could not load the data provided. Make sure you are using the correct version of the XACT tool.
InvalidOperationException	The engine could not be created.

Remarks

The *lookAheadTime* parameter specifies how far ahead the audio engine will look when determining when to transition to another sound. For sounds linked to Play events that use a wave variation list, this parameter should be set to greater than 250 milliseconds. Doing so should give the audio engine enough time to prepare streamed data before it is needed for the loop iteration.

See Also

Reference

[AudioEngine Class](#)











[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetCategory	Gets an audio category.
	GetGlobalVariable	Gets the value of a global variable.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetGlobalVariable	Sets the value of a global variable.
	ToString	(Inherited from Object .)
	Update	Performs periodic work required by the audio engine.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AudioEngine Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
AudioEngine.Dispose ()	Immediately releases the unmanaged resources used by this object.
AudioEngine.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.GetCategory Method

Gets an audio category.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AudioCategory GetCategory (  
    string name  
)
```

Parameters

name

Friendly name of the category to get.

Return Value

Audio category.

Exceptions

Exception type	Condition
InvalidOperationException	This resource could not be created.

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.GetGlobalVariable Method

Gets the value of a global variable.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GetGlobalVariable (  
    string name  
)
```

Parameters

name

Friendly name of the variable.

Return Value

Value of the variable.

Remarks

A global variable has global scope. It can be accessed by all code within a project.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .

See Also

Concepts

[Audio Overview](#)

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.SetGlobalVariable Method

Sets the value of a global variable.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetGlobalVariable (
    string name,
    float value
)
```

Parameters

name

Value of the global variable.

value

Friendly name of the global variable.

Remarks

A global variable has global scope. It can be accessed by all code within a project.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.Update Method

Performs periodic work required by the audio engine.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Update ()
```

Remarks

This method drives the audio engine, and must be called once per frame.

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Change Sound Volume Levels Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[AudioEngine Class](#)



[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	RendererDetails	Gets a collection of audio renderers.

See Also

Reference

[AudioEngine Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine.RendererDetails Property

Gets a collection of audio renderers.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<RendererDetail> RendererDetails { get; }
```

Property Value

Collection of audio renderers.

See Also

Reference

[AudioEngine Class](#)


[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioEngine Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[AudioEngine Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioEngine.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[AudioEngine Class](#)

[AudioEngine Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AudioListener Class

Represents a 3D audio listener. This object, used in combination with an [AudioEmitter](#), can simulate 3D audio effects for a given [Cue](#) or [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class AudioListener
```

See Also

Tasks

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)

Reference

[AudioListener Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





AudioListener Members

The following tables list the members exposed by the AudioListener type.






Public Constructors

	Name	Description
	AudioListener	Initializes a new instance of this class.



Public Properties

	Name	Description
	Forward	Gets or sets the forward orientation vector for this listener.
	Position	Gets or sets the position of this listener.
	Up	Gets or sets the upward orientation vector for this listener.
	Velocity	Gets or sets the velocity vector of this listener.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AudioListener Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioListener Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AudioListener ()
```

See Also

Reference

[AudioListener Class](#)






[AudioListener Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioListener Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[AudioListener Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioListener Properties

Public Properties

	Name	Description
	Forward	Gets or sets the forward orientation vector for this listener.
	Position	Gets or sets the position of this listener.
	Up	Gets or sets the upward orientation vector for this listener.
	Velocity	Gets or sets the velocity vector of this listener.

See Also

Reference

[AudioListener Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

AudioListener.Forward Property

Gets or sets the forward orientation vector for this listener.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Forward { get; set; }
```

Property Value

The forward orientation vector.

Remarks

By default, this value is [Vector3.Forward](#). In most cases, it is not necessary to modify this value.

Doppler values between an [AudioEmitter](#) and an [AudioListener](#) are calculated by the relationship between [AudioListener.Velocity](#) and [AudioEmitter.Velocity](#) with respect to the axes defined by the [Forward](#) and [Up](#) vectors of each.

The values of the [Forward](#) and [Up](#) vectors must be orthonormal (at right angles to one another). Behavior is undefined if these vectors are not orthonormal.

See Also

Reference

[AudioListener Class](#)

[AudioListener Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioListener.Position Property

Gets or sets the position of this listener.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position { get; set; }
```

Property Value

The position vector.

Remarks

By default, the value of this property is [Vector3.Zero](#).

The relative positions of an [AudioEmitter](#) and an [AudioListener](#) are used to determine speaker positioning of a sound.

See Also

Tasks

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)

Reference

[AudioListener Class](#)

[AudioListener Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioListener.Up Property

Gets or sets the upward orientation vector for this listener.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Up { get; set; }
```

Property Value

The upward orientation vector.

Remarks

By default, this value is [Vector3.Up](#). In most cases, it is not necessary to modify this value.

Doppler values between an [AudioEmitter](#) and an [AudioListener](#) are calculated by the relationship between [AudioListener.Velocity](#) and [AudioEmitter.Velocity](#) with respect to the axes defined by the [Forward](#) and [Up](#) vectors of each.

The values of the [Forward](#) and [Up](#) vectors must be orthonormal (at right angles to one another). Behavior is undefined if these vectors are not orthonormal.

See Also

Reference

[AudioListener Class](#)

[AudioListener Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioListener.Velocity Property

Gets or sets the velocity vector of this listener.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Velocity { get; set; }
```

Property Value

The velocity vector.

Remarks

By default, the value of this property is [Vector3.Zero](#).

The Doppler effect value applied to a [Cue](#) is based on the relative [Velocity](#) values of the [AudioEmitter](#) and [AudioListener](#), scaled by the [DopplerScale](#) value.

The [Velocity](#) property is used only to calculate Doppler values. It is not applied to the [Position](#) vector or otherwise used to set game state. You must set [Velocity](#) and [Position](#) each frame to maintain accurate 3D audio values in your game.

See Also

Reference

[AudioListener Class](#)

[AudioListener Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AudioStopOptions Enumeration

Controls how [Cue](#) objects should stop when [Stop](#) is called.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AudioStopOptions
```

Members

Member name	Description
AsAuthored	Indicates the cue should stop normally, playing any release phase or transition specified in the content.
Immediate	Indicates the cue should stop immediately, ignoring any release phase or transition specified in the content.

See Also

Reference

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Cue Class

Defines methods for managing the playback of sounds.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Cue : IDisposable
```

Remarks

Cues are what programmers use to play sounds. Cues are typically played when certain game events occur, such as footsteps or gunshots. A cue is composed of one or more sounds, so when the cue is triggered, the set of associated sounds is heard.

A sound specifies how one or more waves should be played. A sound also has specific properties such as volume and pitch. The sound designer can adjust these properties.

The advantage to using the Audio API to reference cues rather than specific sounds is that an audio designer can reassign sounds to a cue in the sound bank without programmer intervention. For example, a sound designer can try various gunshot waves associated with a particular game event without requiring the programmer to change code or rename sounds.

Cues and sounds are referenced through [SoundBank](#) objects. The waves that compose a sound are referenced through [WaveBank](#) objects.

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[Cue Members](#)










[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista














Cue Members

The following tables list the members exposed by the Cue type.



Public Properties

Name	Description
 IsCreated	Returns whether the cue has been created.
 IsDisposed	Gets a value indicating whether the object has been disposed.
 IsPaused	Returns whether the cue is currently paused.
 IsPlaying	Returns whether the cue is playing.
 IsPrepared	Returns whether the cue is prepared to play.
 IsPreparing	Returns whether the cue is preparing to play.
 IsStopped	Returns whether the cue is currently stopped.
 IsStopping	Returns whether the cue is stopping playback.
 Name	Returns the friendly name of the cue.


Public Methods

Name	Description
 Apply3D	Calculates the 3D audio values between an AudioEmitter and an AudioListener object, and applies the resulting values to this Cue .
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetVariable	Gets a cue-instance variable value based on its friendly name.
 Pause	Pauses playback.
 Play	Requests playback of a prepared or preparing Cue .
 ReferenceEquals	(Inherited from Object .)
 Resume	Resumes playback of a paused Cue .
 SetVariable	Sets the value of a cue-instance variable based on its friendly name.
 Stop	Stops playback of a Cue .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the Cue .
 MemberwiseClone	(Inherited from Object .)

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)














Reference

[Cue Class](#)



[Microsoft.Xna.Framework.Audio Namespace](#)

Cue Methods

Public Methods

Name	Description
 Apply3D	Calculates the 3D audio values between an AudioEmitter and an AudioListener object, and applies the resulting values to this Cue .
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetVariable	Gets a cue-instance variable value based on its friendly name.
 Pause	Pauses playback.
 Play	Requests playback of a prepared or preparing Cue .
 ReferenceEquals	(Inherited from Object .)
 Resume	Resumes playback of a paused Cue .
 SetVariable	Sets the value of a cue-instance variable based on its friendly name.
 Stop	Stops playback of a Cue .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the Cue .
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Cue Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Cue.Apply3D Method

Calculates the 3D audio values between an [AudioEmitter](#) and an [AudioListener](#) object, and applies the resulting values to this [Cue](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Apply3D (
    AudioListener listener,
    AudioEmitter emitter
)
```

Parameters

listener

The listener to calculate.

emitter

The emitter to calculate.

Exceptions

Exception type	Condition
ArgumentNullException	Both <i>listener</i> and <i>emitter</i> must be non- null .
InvalidOperationException	You must call the Cue.Apply3D method before you call the Cue.Play method.

Remarks

If you want to apply 3D effects to a [Cue](#), you must call this method before you call the [Play](#) method. Not doing so will throw an exception the next time [Apply3D](#) is called.

Calling this method automatically sets the speaker mix for any sound played by this cue to a value calculated by the difference in [Position](#) values between *listener* and *emitter*. In preparation for the mix, the sound is converted to monoaural. Any stereo information in the sound is discarded.

Calling this method sets the cue-instance variables *Distance*, *DopplerPitchScalar*, and *OrientationAngle* to the resulting values of the 3D calculation between *listener* and *emitter*. These values do not modify sound attenuation over distance, or pitch shifting via Doppler values, on their own. You must set up a Runtime Parameter Curve (RPC) that defines how to map the cue-instance variable values to pitch and volume shifts, and associate sounds to these curves in the Microsoft Cross-Platform Audio Creation Tool (XACT). For information on doing so, see [How To: Apply Attenuation and Doppler 3D Audio Effects](#).

Example

When first retrieving a [Cue](#) object using [SoundBank.GetCue](#), you must call [Cue.Apply3D](#) before calling [Cue.Play](#), if you want to apply 3D effects to the cue.

C#

```
// 3D audio objects
AudioEmitter emitter = new AudioEmitter();
AudioListener listener = new AudioListener();
Cue cue;

protected override void Initialize()
{
    base.Initialize();

    // Initialize audio objects
    engine = new AudioEngine("Content\\Audio\\3DAudio.xgs");
    soundBank = new SoundBank(engine, "Content\\Audio\\Sound Bank.xsb");
    waveBank = new WaveBank(engine, "Content\\Audio\\Wave Bank.xwb");

    // Get the cue and play it.
    // For 3D cues, you must call Apply3D before calling Play.
    cue = soundBank.GetCue("buzz");
    cue.Apply3D(listener, emitter);
```

```
    cue.Play();  
}
```

Once you have called [Cue.Apply3D](#) then [Cue.Play](#), you can call [Cue.Apply3D](#) each [Game.Update](#) loop after specifying new positions and velocities for the *listener* and *emitter* objects.

C#

```
protected override void Update(GameTime gameTime)  
{  
    // Allow the game to exit.  
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==  
        ButtonState.Pressed)  
        this.Exit();  
  
    // Move the object around in a circle.  
    Vector3 objectPos = new Vector3(  
        (float)Math.Cos(gameTime.TotalGameTime.Seconds) / 2,  
        0,  
        (float)Math.Sin(gameTime.TotalGameTime.Seconds));  
  
    // Apply 3D settings to the cue.  
    emitter.Position = objectPos;  
    cue.Apply3D(listener, emitter);  
  
    // Update the audio engine  
    engine.Update();  
  
    base.Update(gameTime);  
}
```

See Also

Tasks

[How To: Apply Basic 3D Positional Effects to a Cue](#)

[How To: Apply Attenuation and Doppler 3D Audio Effects](#)

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [Cue](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.GetVariable Method

Gets a cue-instance variable value based on its friendly name.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GetVariable (  
    string name  
)
```

Parameters

name

Friendly name of the variable.

Return Value

Value of the variable.

Remarks

Cue-instance variables are useful when multiple instantiations of a single cue (and its associated sounds) are required (for example, a "car" cue where there may be more than one car at any given time). While a global variable allows multiple audio elements to be controlled in unison, a cue instance variable grants discrete control of each instance of a cue, even for each copy of the same cue.

The friendly name is a text string that the designer can associate with the cue when it is built.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.Pause Method

Pauses playback.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Pause ()
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.Play Method

Requests playback of a prepared or preparing [Cue](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Play ()
```

Remarks

Calling **Play** when the [Cue](#) already is playing can result in an [InvalidOperationException](#).

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.Resume Method

Resumes playback of a paused **Cue**.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Resume ()
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.SetVariable Method

Sets the value of a cue-instance variable based on its friendly name.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVariable (  
    string name,  
    float value  
)
```

Parameters

name

Friendly name of the variable to set.

value

Value to assign to the variable.

RemarksThe friendly name is a text string that the designer can associate with the cue when it is built.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

Cue.Stop Method

Stops playback of a [Cue](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Stop (  
    AudioStopOptions options  
)
```

Parameters

options

Enumerated value specifying how the sound should stop. If set to **None**, the sound will play any release phase or transition specified in the audio designer. If set to **Immediate**, the sound will stop immediately, ignoring any release phases or transitions.

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[Cue Class](#)










[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue Properties

Public Properties

	Name	Description
	IsCreated	Returns whether the cue has been created.
	IsDisposed	Gets a value indicating whether the object has been disposed.
	IsPaused	Returns whether the cue is currently paused.
	IsPlaying	Returns whether the cue is playing.
	IsPrepared	Returns whether the cue is prepared to play.
	IsPreparing	Returns whether the cue is preparing to play.
	IsStopped	Returns whether the cue is currently stopped.
	IsStopping	Returns whether the cue is stopping playback.
	Name	Returns the friendly name of the cue.

See Also

Reference

[Cue Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Cue.IsCreated Property

Returns whether the cue has been created.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsCreated { get; }
```

Property Value

true if the cue is created; **false** otherwise.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsDisposed Property

Gets a value indicating whether the object has been disposed.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object has been disposed; **false** otherwise.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsPaused Property

Returns whether the cue is currently paused.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPaused { get; }
```

Property Value

true if the cue is paused; **false** otherwise.

Remarks

[IsPlaying](#) and [IsPaused](#) both return **true** if a cue is paused while playing.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsPlaying Property

Returns whether the cue is playing.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPlaying { get; }
```

Property Value

true if the cue is playing; **false** otherwise.

Remarks

[IsPlaying](#) and [IsPaused](#) both return **true** if a cue is paused while playing.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsPrepared Property

Returns whether the cue is prepared to play.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPrepared { get; }
```

Property Value

true if the cue is prepared to play; **false** otherwise.

Remarks

This property returns **true** only if the cue is prepared but has not yet been played. Cues that are played return **true** for [IsPlaying](#) if they are currently playing or **true** for [IsStopped](#) if they are stopped.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsPreparing Property

Returns whether the cue is preparing to play.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPreparing { get; }
```

Property Value

true if the cue is preparing to play; **false** otherwise.

This property returns **true** only if the cue is preparing but has not yet been prepared. Cues that have been prepared return **true** for [IsPrepared](#).

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsStopped Property

Returns whether the cue is currently stopped.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsStopped { get; }
```

Property Value

true if the cue is stopped; **false** if otherwise.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.IsStopping Property

Returns whether the cue is stopping playback.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsStopping { get; }
```

Property Value

true if the cue is stopping; **false** if otherwise.

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue.Name Property

Returns the friendly name of the cue.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

Friendly name of the cue.

Remarks The friendly name is a text string that the designer can associate with the cue when it is built.

See Also

Reference

[Cue Class](#)


[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Cue Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[Cue Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Cue.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[Cue Class](#)

[Cue Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

InstancePlayLimitException Class

The exception that is thrown when there is an attempt to play more than 16 [SoundEffectInstance](#) sounds concurrently.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class InstancePlayLimitException : ExternalException
```

Remarks

Only 16 [SoundEffectInstance](#) sounds can be playing at one time. If an attempt is made to play more than 16, this exception is thrown. Paused or stopped [SoundEffectInstance](#) objects do not count against this limit.

See Also

Reference

[InstancePlayLimitException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

InstancePlayLimitException Members









The exception that is thrown when there is an attempt to play more than 16 [SoundEffectInstance](#) sounds concurrently.

The following tables list the members exposed by the InstancePlayLimitException type.


Public Constructors

Name	Description
 InstancePlayLimitException	Overloaded. Initializes a new instance of the InstancePlayLimitException class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[InstancePlayLimitException Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

InstancePlayLimitException Constructor

Initializes a new instance of the [InstancePlayLimitException](#) class.

Overload List

Name	Description
InstancePlayLimitException ()	Initializes a new instance of the InstancePlayLimitException class.
InstancePlayLimitException (String)	Initializes a new instance of the InstancePlayLimitException class with a specified error message.
InstancePlayLimitException (String, Exception)	Initializes a new instance of the InstancePlayLimitException class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[InstancePlayLimitException Class](#)

[InstancePlayLimitException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

InstancePlayLimitException Constructor ()

Initializes a new instance of the [InstancePlayLimitException](#) class.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public InstancePlayLimitException ()
```

See Also

Reference

[InstancePlayLimitException Class](#)

[InstancePlayLimitException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

InstancePlayLimitException Constructor (String)

Initializes a new instance of the [InstancePlayLimitException](#) class with a specified error message.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public InstancePlayLimitException (  
    string message  
)
```

Parameters

message

A [String](#) that describes the error. The content of *message* is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

Remarks

The [Message](#) property of the new instance is initialized to *message*.

The [InnerException](#) property of the new instance is initialized to **null**.

See Also

Reference

[InstancePlayLimitException Class](#)

[InstancePlayLimitException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

InstancePlayLimitException Constructor (String, Exception)

Initializes a new instance of the [InstancePlayLimitException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public InstancePlayLimitException (  
    string message,  
    Exception inner  
)
```

Parameters

message

Error message that explains the reason for the exception.

inner

[Exception](#) that is the cause of the current exception. If *innerException* is not **null**, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[InstancePlayLimitException Class](#)







[InstancePlayLimitException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

InstancePlayLimitException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[InstancePlayLimitException Class](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

InstancePlayLimitException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[InstancePlayLimitException Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

NoAudioHardwareException Class

The exception that is thrown when no audio hardware is present, or when audio hardware is installed, but the device drivers for the audio hardware are not present or enabled.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class NoAudioHardwareException : ExternalException
```

See Also

Reference

[NoAudioHardwareException Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









NoAudioHardwareException Members

The following tables list the members exposed by the NoAudioHardwareException type.

Public Constructors

Name	Description
 NoAudioHardwareException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NoAudioHardwareException Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

NoAudioHardwareException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
NoAudioHardwareException ()	Initializes a new instance of this class.
NoAudioHardwareException (String)	Initializes a new instance of this class with a specified error message.
NoAudioHardwareException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[NoAudioHardwareException Class](#)

[NoAudioHardwareException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

NoAudioHardwareException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NoAudioHardwareException ()
```

See Also

Reference

[NoAudioHardwareException Class](#)

[NoAudioHardwareException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NoAudioHardwareException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NoAudioHardwareException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[NoAudioHardwareException Class](#)

[NoAudioHardwareException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NoAudioHardwareException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NoAudioHardwareException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[NoAudioHardwareException Class](#)







[NoAudioHardwareException Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NoAudioHardwareException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[NoAudioHardwareException Class](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

NoAudioHardwareException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[NoAudioHardwareException Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

RendererDetail Structure

Represents an audio renderer, which is a device that can render audio to a user.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct RendererDetail
```

See Also

Reference

[RendererDetail Members](#)



[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








RendererDetail Members

The following tables list the members exposed by the `RendererDetail` type.



Public Properties

	Name	Description
	FriendlyName	Gets the human-readable name for the renderer.
	RendererId	Specifies the string that identifies the renderer.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares two objects to determine whether they are the same.
	Op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[RendererDetail Structure](#)



[Microsoft.Xna.Framework.Audio Namespace](#)

RendererDetail Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[RendererDetail Structure](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

RendererDetail.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
RendererDetail.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
RendererDetail.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

RendererDetail.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object to compare to this object.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RendererDetail.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RendererDetail.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    RendererDetail left,  
    RendererDetail right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RendererDetail.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    RendererDetail left,  
    RendererDetail right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RendererDetail.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[RendererDetail Structure](#)



[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RendererDetail Properties

Public Properties

	Name	Description
	FriendlyName	Gets the human-readable name for the renderer.
	RendererId	Specifies the string that identifies the renderer.

See Also

Reference

[RendererDetail Structure](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

RendererDetail.FriendlyName Property

Gets the human-readable name for the renderer.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string FriendlyName { get; }
```

Property Value

The name of the renderer.

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RendererDetail.RendererId Property

Specifies the string that identifies the renderer.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string RendererId { get; }
```

Property Value

A string identifying the renderer. If this value is **null**, the default audio output device is used.

Caution

On Xbox 360, this property is unused and ignored.

See Also

Reference

[RendererDetail Structure](#)

[RendererDetail Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank Class

Represents a sound bank, which is a collection of cues.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class SoundBank : IDisposable
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[SoundBank Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista



SoundBank Members

The following tables list the members exposed by the SoundBank type.









Public Constructors

Name	Description
 SoundBank	Initializes a new instance of this class using a sound bank from file.



Public Properties

Name	Description
 IsDisposed	Gets a value that indicates whether the object is disposed.
 IsInUse	Returns whether the sound bank is currently in use.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetCue	Gets a cue from the sound bank.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 PlayCue	Overloaded. Plays a cue.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[SoundBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundBank Constructor

Initializes a new instance of this class using a sound bank from file.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundBank (  
    AudioEngine audioEngine,  
    string filename  
)
```

Parameters

audioEngine

Audio engine that will be associated with this sound bank.

filename

Path to the sound bank file.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>audioEngine</i> or <i>filename</i> parameter is null .
ArgumentException	XACT could not load the data provided. Make sure you are using the correct version of the XACT tool.
InvalidOperationException	Unable to create the SoundBank resource.

See Also

Reference

[SoundBank Class](#)









[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetCue	Gets a cue from the sound bank.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	PlayCue	Overloaded. Plays a cue.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SoundBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundBank.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
SoundBank.Dispose ()	Immediately releases the unmanaged resources used by this object.
SoundBank.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundBank.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank.GetCue Method

Gets a cue from the sound bank.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Cue GetCue (
    string name
)
```

Parameters

name

Friendly name of the cue to get.

Return Value

Cue object.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .

Remarks

Each [Cue](#) instance that you retrieve is unique, even when retrieving multiple cues with the same name. This allows multiple instances of the same [Cue](#) to exist and play simultaneously.

The [Cue](#) instance that you retrieve using this method is valid only as long as it is in scope. If the [Cue](#) instance is allowed to go out of scope, such as in the case of the [Cue](#) being retrieved and played without otherwise being stored, the [Cue](#) is marked to be disposed by the garbage collector. This will result in the [Cue](#) stopping playback at an indeterminate time when the garbage collector disposes of the [Cue](#). To prevent this behavior, store [Cue](#) objects to prevent them from falling out of scope. Alternatively, you can call [PlayCue](#) to play a cue immediately that will not fall out of scope.

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank.PlayCue Method

Plays a cue.

Overload List

Name	Description
SoundBank.PlayCue (String)	Plays a cue.
SoundBank.PlayCue (String, AudioListener, AudioEmitter)	Plays a cue using 3D positional information specified in an AudioListener and AudioEmitter .

See Also

Tasks

[How To: Play a Sound Using XACT](#)

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundBank.PlayCue Method (String)

Plays a cue.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void PlayCue (  
    string name  
)
```

Parameters

name

Name of the cue to play.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .
InvalidOperationException	The <i>name</i> parameter specifies a cue that does not exist.

Remarks

Each [Cue](#) instance that you play is unique, even when playing multiple cues with the same name. This allows multiple instances of the same [Cue](#) to play simultaneously.

Calling this method does not return a [Cue](#) object. Therefore, once the cue has started playing, it cannot be controlled through any methods of [Cue](#), such as [Stop](#), [Pause](#), or [Resume](#). The cue will play until its authored end, at which point it will be released.

Because no [Cue](#) object is retrieved, cues played using this method are also unable to use [Apply3D](#) to apply 3D effects to the cue. To retrieve a cue that you can control during playback, or apply 3D effects to, use [GetCue](#).

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Reference

[SoundBank.PlayCue Method](#)

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank.PlayCue Method (String, AudioListener, AudioEmitter)

Plays a cue using 3D positional information specified in an [AudioListener](#) and [AudioEmitter](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void PlayCue (
    string name,
    AudioListener listener,
    AudioEmitter emitter
)
```

Parameters

name

Name of the cue to play.

listener

[AudioListener](#) that specifies listener 3D audio information.

emitter

[AudioEmitter](#) that specifies emitter 3D audio information.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>name</i> parameter is null .
InvalidOperationException	The <i>name</i> parameter specifies a cue that does not exist.

Remarks

The 3D audio information in the specified *listener* and *emitter* is used to initialize the cue instance and apply a static positional 3D audio effect (the direction from which the player hears the sound). Position and velocity are *not* updated over time; the 3D audio effects are calculated based on the emitter and listener position at the time of the call. Typically, this is used for short-lived sounds, or when the object and listener are stationary for the duration of the sound but placed somewhere in the world space.

For example, a sound that is played off to the right will continue to sound as if it is coming from the right, even if the listener moves or turns. For very short sounds, this may be imperceptible and thus acceptable. As another example, a voice that comes from the player's vehicle will always be the same distance and direction relative to the player, so static positioning in this case would be correct.

Attenuation and doppler effects, if enabled, will also be static for the duration of the sound.

To enable dynamic 3D audio when the emitter or listener are moving, use [GetCue](#) and [Apply3D](#) instead.

Each [Cue](#) instance that you play is unique, even when playing multiple cues with the same name. This allows multiple instances of the same [Cue](#) to play simultaneously.

Calling this method does not return a [Cue](#) object. Therefore, once the cue has started playing, it cannot be controlled through any methods of [Cue](#), such as [Stop](#), [Pause](#), or [Resume](#). The cue will play until its authored end, at which point it will be released.

Because no [Cue](#) object is retrieved, cues played using this method are also unable to use [Apply3D](#) to apply dynamic 3D effects to the cue. To retrieve a cue that you can control during playback, or apply dynamic 3D effects to, use [GetCue](#).

For more information about 3D audio, see [How To: Apply Basic 3D Positional Effects to a Cue](#) and [How To: Apply Attenuation and Doppler 3D Audio Effects](#).

See Also

Concepts

[Audio Overview](#)

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

[How To: Apply Basic 3D Positional Effects to a Cue](#)

[How To: Apply Attenuation and Doppler 3D Audio Effects](#)

Reference

[SoundBank.PlayCue Method](#)

[SoundBank Class](#)



[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	IsInUse	Returns whether the sound bank is currently in use.

See Also

Reference

[SoundBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundBank.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank.IsInUse Property

Returns whether the sound bank is currently in use.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsInUse { get; }
```

Property Value

true if the sound bank is in use; **false** otherwise.

Remarks

This value will return **true** if the sound bank is currently referenced by at least one valid cue instance or other client. For example, the game itself might reference the sound bank.

See Also

Reference

[SoundBank Class](#)


[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundBank Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[SoundBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundBank.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[SoundBank Class](#)

[SoundBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundEffect Class

Provides a loaded sound resource. You can play multiple instances of the **SoundEffect** by calling [Play](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SoundEffect : IDisposable
```

Remarks

A **SoundEffect** contains the audio data and metadata (such as wave data and loop information) loaded from a sound file. You can create multiple [SoundEffectInstance](#) objects, and play them from a single **SoundEffect**. These objects share the resources of that **SoundEffect**.

You can create a **SoundEffect** by calling [ContentManager.Load](#). When you make that call, use the type **SoundEffect** and the asset name of an audio file. The audio file must be part of the **Content** project. Be sure to use the **SoundEffect - XNA Framework** content processor.

The only limit to the number of loaded **SoundEffect** objects is memory. A loaded **SoundEffect** will continue to hold its memory resources throughout its lifetime. All [SoundEffectInstance](#) objects created from a **SoundEffect** share memory resources. When a **SoundEffect** object is destroyed, all [SoundEffectInstance](#) objects previously created by that **SoundEffect** will stop playing and become invalid.

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[How To: Play a Sound](#)

[How To: Loop a Sound](#)

[How To: Change the Pitch or Volume of a Sound](#)

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Reference

[SoundEffectInstance Class](#)

[ContentManager.Load Generic Method](#)

[SoundEffect Members](#)








[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










SoundEffect Members

The following tables list the members exposed by the SoundEffect type.



Public Properties

	Name	Description
	S DistanceScale	Gets or sets a value that adjusts the effect of distance calculations on the sound (emitter).
	S DopplerScale	Gets or sets a value that adjusts the effect of doppler calculations on the sound (emitter).
	Duration	Gets the duration of the SoundEffect .
	IsDisposed	Gets a value that indicates whether the object is disposed.
	S MasterVolume	Gets or sets the master volume that affects all SoundEffectInstance sounds.
	Name	Gets or sets the asset name of the SoundEffect .
	S SpeedOfSound	Returns the speed of sound: 343.5 meters per second.

Public Methods

	Name	Description
	CreateInstance	Creates a new SoundEffectInstance for this SoundEffect .
	Dispose	Releases the resources used by the SoundEffect .
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Play	Overloaded. Plays this sound.
	Play3D	Overloaded. Returns a playing SoundEffectInstance (XNA Game Studio 3.0 only).
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the SoundEffect is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also










Reference

[SoundEffect Class](#)



[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffect Methods

Public Methods

	Name	Description
	CreateInstance	Creates a new SoundEffectInstance for this SoundEffect .
	Dispose	Releases the resources used by the SoundEffect .
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Play	Overloaded. Plays this sound.
	Play3D	Overloaded. Returns a playing SoundEffectInstance (XNA Game Studio 3.0 only).
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the SoundEffect is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SoundEffect Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffect.CreateInstance Method

Creates a new [SoundEffectInstance](#) for this [SoundEffect](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundEffectInstance CreateInstance ()
```

Return Value

A new [SoundEffectInstance](#) for this [SoundEffect](#).

Remarks

Creating a [SoundEffectInstance](#) before calling [Play](#) allows you to access advanced playback features, such as 3D positioning.

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[How To: Loop a Sound](#)

[How To: Change the Pitch or Volume of a Sound](#)

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Dispose Method

Releases the resources used by the [SoundEffect](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call **Dispose** only when you are finished using the **SoundEffect**. The **Dispose** method leaves the **SoundEffect** in an unusable state. After calling **Dispose**, you must release all references to the **SoundEffect** in order for the garbage collector to reclaim the memory that the **SoundEffect** was occupying. For more information, see [Cleaning Up Unmanaged Resources](#) and [Implementing a Dispose Method](#).

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [SoundEffect](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Play Method

Plays this sound.

Overload List

Name	Description
SoundEffect.Play ()	Creates a new SoundEffectInstance , and plays it.
SoundEffect.Play (Single)	Creates a new SoundEffectInstance , and plays it at the specified volume (XNA Game Studio 3.0 only).
SoundEffect.Play (Single, Single, Single)	Creates a new SoundEffectInstance , and plays it using the specified volume, pitch, and panning.
SoundEffect.Play (Single, Single, Single, Boolean)	Creates a new SoundEffectInstance , and plays it using the specified volume, pitch, panning, and loop control (XNA Game Studio 3.0 only).

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[How To: Play a Sound](#)

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffect.Play Method ()

Creates a new [SoundEffectInstance](#), and plays it.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Play ()
```

Return Value

true if the sound is playing back successfully; otherwise, **false**.

Remarks

Play will return **false** if there are too many sounds already playing.

To loop a sound or apply 3D effects, call [CreateInstance](#) instead of **Play**, and [SoundEffectInstance.Play](#).

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[How To: Play a Sound](#)

[How To: Loop a Sound](#)

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Play Method (Single)

Creates a new [SoundEffectInstance](#), and plays it at the specified volume (XNA Game Studio 3.0 only).

⚠ Caution

This method is available only when developing for XNA Game Studio 3.0.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundEffectInstance Play (  
    float volume  
)
```

Parameters

volume

Volume, ranging from 0.0f (silence) to 1.0f (full volume). 1.0f is full volume relative to [SoundEffect.MasterVolume](#).

Return Value

A new, playing, [SoundEffectInstance](#).

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundEffect.Play Method (Single, Single, Single)

Creates a new [SoundEffectInstance](#), and plays it using the specified volume, pitch, and panning.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Play (  
    float volume,  
    float pitch,  
    float pan  
)
```

Parameters

volume

Volume, ranging from 0.0f (silence) to 1.0f (full volume). 1.0f is full volume relative to [SoundEffect.MasterVolume](#).

pitch

Pitch adjustment, ranging from -1.0f (down one octave) to 1.0f (up one octave). 0.0f is unity (normal) pitch.

pan

Panning, ranging from -1.0f (full left) to 1.0f (full right). 0.0f is centered.

Return Value

true if the sound is playing back successfully; otherwise, **false**.

Remarks

Play will return **false** if there are too many sounds already playing.

To loop a sound or apply 3D effects, call [CreateInstance](#) instead of **Play**, and [SoundEffectInstance.Play](#).

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[How To: Play a Sound](#)

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Play Method (Single, Single, Single, Boolean)

Creates a new [SoundEffectInstance](#), and plays it using the specified volume, pitch, panning, and loop control (XNA Game Studio 3.0 only).

⚠Caution

This method is available only when developing for XNA Game Studio 3.0. For XNA Game Studio 3.1 or later, please use [SoundEffectInstance.IsLooped](#). For more information, please see [How To: Loop a Sound](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundEffectInstance Play (  
    float volume,  
    float pitch,  
    float pan,  
    bool loop  
)
```

Parameters

volume

Volume, ranging from 0.0f (silence) to 1.0f (full volume). 1.0f is full volume relative to [SoundEffect.MasterVolume](#).

pitch

Pitch adjustment, ranging from -1.0f (down one octave) to 1.0f (up one octave). 0.0f is unity (normal) pitch.

pan

Panning, ranging from -1.0f (full left) to 1.0f (full right). 0.0f is centered.

loop

Whether to loop the sound indefinitely, until stopped by the application. Specify **true** to continuously loop the sound's loop region, or **false** to play the entire sound just once.

Return Value

A new, playing, [SoundEffectInstance](#).

Remarks

If *loop* is **true**, the sound's loop region will play continuously until stopped with a call to [Stop](#) or [Pause](#) on the returned [SoundEffectInstance](#). The loop region as authored in the sound file is used. If the sound file has no loop region, the entire sound is looped.

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundEffect.Play3D Method

Returns a playing [SoundEffectInstance](#) (XNA Game Studio 3.0 only).

⚠Caution

This method is available only when developing for XNA Game Studio 3.0. For XNA Game Studio 3.1 or later, please use [SoundEffectInstance.Apply3D](#). For more information, please see [How To: Apply Basic 3D Positional Effects to a SoundEffect](#).

🔍Zune Specific Information

3D audio is not supported. All 3D audio effects are mapped to the volume and pan properties used by [SoundEffectInstance Properties](#).

Overload List

Name	Description
SoundEffect.Play3D (AudioListener, AudioEmitter)	Returns a playing SoundEffectInstance at the specified position with the default volume and pitch (XNA Game Studio 3.0 only).
SoundEffect.Play3D (AudioListener, AudioEmitter, Single, Single, Boolean)	Returns a playing SoundEffectInstance at the specified position using the specified values for volume, pitch, and looping (XNA Game Studio 3.0 only).
SoundEffect.Play3D (AudioListener[], AudioEmitter, Single, Single, Boolean)	Returns a playing SoundEffectInstance at the specified positions using the specified values for volume, pitch, and looping (XNA Game Studio 3.0 only).

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffect.Play3D Method (AudioListener, AudioEmitter)

Returns a playing [SoundEffectInstance](#) at the specified position with the default volume and pitch (XNA Game Studio 3.0 only).

⚠ Caution

This method is available only when developing for XNA Game Studio 3.0. For XNA Game Studio 3.1 or later, please use [SoundEffectInstance.Apply3D](#). For more information, please see [How To: Apply Basic 3D Positional Effects to a SoundEffect](#).

🔧 Zune Specific Information

3D audio is not supported. All 3D audio effects are mapped to the volume and pan properties used by [SoundEffectInstance Properties](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundEffectInstance Play3D (  
    AudioListener listener,  
    AudioEmitter emitter  
)
```

Parameters

listener

Listener position.

emitter

Position of the emitter.

Return Value

A playing [SoundEffectInstance](#) at the specified position with the default volume and pitch.

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundEffect.Play3D Method (AudioListener, AudioEmitter, Single, Single, Boolean)

Returns a playing [SoundEffectInstance](#) at the specified position using the specified values for volume, pitch, and looping (XNA Game Studio 3.0 only).

⚠Caution

This method is available only when developing for XNA Game Studio 3.0. For XNA Game Studio 3.1 or later, please use [SoundEffectInstance.Apply3D](#). For more information, please see [How To: Apply Basic 3D Positional Effects to a SoundEffect](#).

📌Zune Specific Information

3D audio is not supported. All 3D audio effects are mapped to the volume and pan properties used by [SoundEffectInstance Properties](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundEffectInstance Play3D (  
    AudioListener listener,  
    AudioEmitter emitter,  
    float volume,  
    float pitch,  
    bool loop  
)
```

Parameters

listener

Listener position.

emitter

Position of the emitter.

volume

Volume to use when beginning playback. Ranging from 0.0f (silence) to 1.0f (full volume). 1.0f is full volume relative to [SoundEffect.MasterVolume](#).

pitch

Pitch adjustment, ranging from -1.0f (down one octave) to 1.0f (up one octave). 0.0f is unity (normal) pitch.

loop

Whether to loop the sound indefinitely, until stopped by the application. Specify **true** to continuously loop the sound's loop region, or **false** to play the entire sound just once.

Return Value

A playing [SoundEffectInstance](#) at the specified position using the specified values for volume, pitch, and looping.

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundEffect.Play3D Method (AudioListener[], AudioEmitter, Single, Single, Boolean)

Returns a playing [SoundEffectInstance](#) at the specified positions using the specified values for volume, pitch, and looping (XNA Game Studio 3.0 only).

⚠Caution

This method is available only when developing for XNA Game Studio 3.0. For XNA Game Studio 3.1 or later, please use [SoundEffectInstance.Apply3D](#). For more information, please see [How To: Apply Basic 3D Positional Effects to a SoundEffect](#).

📌Zune Specific Information

3D audio is not supported. All 3D audio effects are mapped to the volume and pan properties used by [SoundEffectInstance Properties](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundEffectInstance Play3D (  
    AudioListener[] listeners,  
    AudioEmitter emitter,  
    float volume,  
    float pitch,  
    bool loop  
)
```

Parameters

listeners

Positions of each listener.

emitter

Position of the emitter.

volume

Volume to use when beginning playback. Ranging from 0.0f (silence) to 1.0f (full volume). 1.0f is full volume relative to [SoundEffect.MasterVolume](#).

pitch

Pitch adjustment, ranging from -1.0f (down one octave) to 1.0f (up one octave). 0.0f is unity (normal) pitch.

loop

Whether to loop the sound indefinitely, until stopped by the application. Specify **true** to continuously loop the sound's loop region, or **false** to play the entire sound just once.

Return Value

A playing [SoundEffectInstance](#) at the specified positions using the specified values for volume, pitch, and looping.

Remarks

Use this method when there is more than one [AudioListener](#) listening. For example, this can be useful for split screen games when there are two players in the game world at the same time.

See Also

Reference

[SoundEffect Class](#)








[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SoundEffect Properties

Public Properties

	Name	Description
 S	DistanceScale	Gets or sets a value that adjusts the effect of distance calculations on the sound (emitter).
 S	DopplerScale	Gets or sets a value that adjusts the effect of doppler calculations on the sound (emitter).
	Duration	Gets the duration of the SoundEffect .
	IsDisposed	Gets a value that indicates whether the object is disposed.
 S	MasterVolume	Gets or sets the master volume that affects all SoundEffectInstance sounds.
	Name	Gets or sets the asset name of the SoundEffect .
 S	SpeedOfSound	Returns the speed of sound: 343.5 meters per second.

See Also

Reference

[SoundEffect Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffect.DistanceScale Property

Gets or sets a value that adjusts the effect of distance calculations on the sound (emitter).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float DistanceScale { get; set; }
```

Property Value

A value that adjustments the effect of Distance and Doppler calculations on the sound (emitter).

Remarks

If sounds are attenuating too fast, which means the sounds get quiet too quickly as they move away from the listener, you need to increase the DistanceScale. If sounds are not attenuating fast enough, decrease the DistanceScale.

Adjustment	Effect
DistanceScale = 0.0	Invalid (DistanceScale will be reset to default of 1.0)
0.0 < DistanceScale <= 1.0	Increases the effect of distance attenuation
DistanceScale = 1.0	No scaling (this is the default setting)
DistanceScale > 1.0	Decrease the effect of distance attenuation

Distance attenuation is calculated using the inverse square law:

$$\text{volume} = 1.0 / \text{distance}$$

Adding DistanceScale into the equations give:

```
If (distance > DistanceScale)
volume = DistanceScale / distance
else
volume = 1.0
```

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.DopplerScale Property

Gets or sets a value that adjusts the effect of doppler calculations on the sound (emitter).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float DopplerScale { get; set; }
```

Property Value

A value that adjusts the effect of doppler calculations on the sound (emitter).

Remarks

DopplerScale changes the relative velocities of emitters and listeners.

If sounds are shifting (pitch) too much for the given relative velocity of the emitter and listener, decrease the DopplerScale. If sounds are not shifting enough for the given relative velocity of the emitter and listener, increase the DopplerScale.

Adjustment	Effect
DopplerScale = 0	No Doppler will be applied
$0.0 < \text{DopplerScale} \leq 1.0$	Decreases the effect of Doppler
DopplerScale = 1.0	No scaling (default)
DopplerScale > 1.0	Increase the effect of Doppler

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Duration Property

Gets the duration of the [SoundEffect](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

A [TimeSpan](#) that represents the duration of the [SoundEffect](#).

Remarks

This is a read-only property. The sound duration is initialized from the sound file and cannot be changed. For information about authoring and adding sound files, see [How To: Add a Sound File to Your Game Using XACT](#).

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.MasterVolume Property

Gets or sets the master volume that affects all [SoundEffectInstance](#) sounds.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float MasterVolume { get; set; }
```

Property Value

Volume, ranging from 0.0f (silence) to 1.0f (current device volume). 1.0f is full volume relative to the current device volume.

Remarks

The sound effect master volume affects all sound effect instances, including currently playing instances and newly created instances. Each [SoundEffectInstance](#) also has its own volume ([SoundEffectInstance.Volume](#)) that is relative to the master volume.

You can adjust the volume of all sounds effects by changing **MasterVolume**. You can adjust the volume of an individual [SoundEffectInstance](#) by changing [SoundEffectInstance.Volume](#), or by specifying an instance volume when you call [Play](#) to create the instance.

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.Name Property

Gets or sets the asset name of the [SoundEffect](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

Asset name of the [SoundEffect](#).

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffect.SpeedOfSound Property

Returns the speed of sound: 343.5 meters per second.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float SpeedOfSound { get; set; }
```

Property Value

The speed of sound: 343.5 meters per second.

Remarks

Use this value to simulate different environments. A smaller speed of sound will exaggerate the doppler effect. A higher speed of sound will reduce the doppler effect. Speed of sound has no impact on distance attenuation.

See Also

Reference

[SoundEffect Class](#)

[SoundEffect Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance Class

Provides a single playing, paused, or stopped instance of a [SoundEffect](#) sound.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SoundEffectInstance : IDisposable
```

Remarks

You can create a **SoundEffectInstance** by calling [SoundEffect.CreateInstance](#). Initially, the **SoundEffectInstance** is created as stopped, but you can play it by calling [Play](#).

You can modify the volume, panning, and pitch of the **SoundEffectInstance** by setting the [Volume](#), [Pitch](#), and [Pan](#) properties.

On Zune, a game can have a maximum of 16 total playing [SoundEffectInstance](#) instances at one time, combined across *all* loaded [SoundEffect](#) objects. The only limit to the total number of loaded **SoundEffectInstance** and **SoundEffect** objects is available memory. However, the user can play only 16 sound effects at one time. Attempts to play a **SoundEffectInstance** beyond this limit will fail. On Windows, there is no hard limit. Playing too many instances can lead to performance degradation. On Xbox 360, the limit is 300 sound effect instances loaded or playing. Dispose of old instances if you need more.

Note

Unlike other platforms, the Xbox 360 has a limit to how many sound effect instances can exist in memory at one time.

See Also

Reference

[SoundEffect.Play Method](#)

[SoundEffectInstance Members](#)







[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune












SoundEffectInstance Members

The following tables list the members exposed by the SoundEffectInstance type.



Public Properties

Name	Description
 IsDisposed	Gets a value that indicates whether the object is disposed.
 IsLooped	Gets a value that indicates whether looping is enabled for the SoundEffectInstance .
 Pan	Gets or sets the panning for the SoundEffectInstance .
 Pitch	Gets or sets the pitch adjustment for the SoundEffectInstance .
 State	Gets the current state (playing, paused, or stopped) of the SoundEffectInstance .
 Volume	Gets or sets the volume of the SoundEffectInstance .

Public Methods

Name	Description
 Apply3D	Overloaded. Applies 3D position to the sound.
 Dispose	Releases the resources used by the Dispose.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Pause	Pauses a SoundEffectInstance .
 Play	Plays or resumes a SoundEffectInstance .
 ReferenceEquals	(Inherited from Object .)
 Resume	Resumes playback for a SoundEffectInstance .
 Stop	Overloaded. Stops playing a SoundEffectInstance .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before the SoundEffectInstance is reclaimed by garbage collection.
 MemberwiseClone	(Inherited from Object .)

See Also












Reference

[SoundEffectInstance Class](#)



[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffectInstance Methods

Public Methods

	Name	Description
	Apply3D	Overloaded. Applies 3D position to the sound.
	Dispose	Releases the resources used by the Dispose.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Pause	Pauses a SoundEffectInstance .
	Play	Plays or resumes a SoundEffectInstance .
	ReferenceEquals	(Inherited from Object .)
	Resume	Resumes playback for a SoundEffectInstance .
	Stop	Overloaded. Stops playing a SoundEffectInstance .
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the SoundEffectInstance is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SoundEffectInstance Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffectInstance.Apply3D Method

Applies 3D position to the sound.

Overload List

Name	Description
SoundEffectInstance.Apply3D (AudioListener, AudioEmitter)	Applies 3D positioning to the sound using a single listener.
SoundEffectInstance.Apply3D (AudioListener[], AudioEmitter)	Applies 3D position to the sound using multiple listeners.

See Also

Tasks

[How To: Play a Sound](#)

[How To: Loop a Sound](#)

[How To: Change the Pitch or Volume of a Sound](#)

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffectInstance.Apply3D Method (AudioListener, AudioEmitter)

Applies 3D positioning to the sound using a single listener.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Apply3D (  
    AudioListener listener,  
    AudioEmitter emitter  
)
```

Parameters

listener

Position of the listener.

emitter

Position of the emitter.

See Also

Tasks

[How To: Play a Sound](#)

[How To: Loop a Sound](#)

[How To: Change the Pitch or Volume of a Sound](#)

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Apply3D Method (AudioListener[], AudioEmitter)

Applies 3D position to the sound using multiple listeners.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Apply3D (  
    AudioListener[] listeners,  
    AudioEmitter emitter  
)
```

Parameters

listeners

Positions of each listener.

emitter

Position of the emitter.

See Also

Tasks

[How To: Play a Sound](#)

[How To: Loop a Sound](#)

[How To: Change the Pitch or Volume of a Sound](#)

[How To: Apply Basic 3D Positional Effects to a SoundEffect](#)

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Dispose Method

Releases the resources used by the **Dispose**.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call **Dispose** only when you are finished using the **SoundEffectInstance**. The **Dispose** method leaves the **SoundEffectInstance** in an unusable state. After calling **Dispose**, you must release all references to the **SoundEffectInstance** in order for the garbage collector to reclaim the memory that the **SoundEffectInstance** was occupying. For more information, see [Cleaning Up Unmanaged Resources](#) and [Implementing a Dispose Method](#).

See Also

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [SoundEffectInstance](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Pause Method

Pauses a [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Pause ()
```

Remarks

To resume a paused [SoundEffectInstance](#), call [Play](#).

See Also

Reference

[SoundEffectInstance.Play Method](#)

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Play Method

Plays or resumes a [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Play ()
```

Remarks

If the [SoundEffectInstance](#) is paused, **Play** resumes playing it at the last played position. If the **SoundEffectInstance** is stopped, **Play** begins to play it.

On Zune, a game can have a maximum of 16 total playing **SoundEffectInstance** instances at one time, combined across *all* loaded [SoundEffect](#) objects. The only limit to the total number of loaded **SoundEffectInstance** and **SoundEffect** objects is available memory. However, the user can play only 16 sound effects at one time. Attempts to play a **SoundEffectInstance** beyond this limit will fail. On Windows, there is no hard limit. Playing too many instances can lead to performance degradation. On Xbox 360, the limit is 300 sound effect instances loaded *or* playing. Dispose of old instances if you need more.

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Reference

[SoundEffectInstance.Stop Method](#)

[SoundEffectInstance.Pause Method](#)

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Resume Method

Resumes playback for a [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Resume ()
```

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Stop Method

Stops playing a [SoundEffectInstance](#).

Overload List

Name	Description
SoundEffectInstance.Stop ()	Immediately stops playing a SoundEffectInstance .
SoundEffectInstance.Stop (Boolean)	Stops playing a SoundEffectInstance , either immediately or as authored.

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Reference

[SoundEffectInstance.Play Method](#)

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffectInstance.Stop Method ()

Immediately stops playing a [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Stop ()
```

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Reference

[SoundEffectInstance.Play Method](#)

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Stop Method (Boolean)

Stops playing a [SoundEffectInstance](#), either immediately or as authored.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Stop (  
    bool immediate  
)
```

Parameters

immediate

Whether to stop playing immediately, or to break out of the loop region and play the release. Specify **true** to stop playing immediately, or **false** to break out of the loop region and play the release phase (the remainder of the sound).

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Reference

[SoundEffectInstance.Play Method](#)

[SoundEffectInstance Class](#)







[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	IsLooped	Gets a value that indicates whether looping is enabled for the SoundEffectInstance .
	Pan	Gets or sets the panning for the SoundEffectInstance .
	Pitch	Gets or sets the pitch adjustment for the SoundEffectInstance .
	State	Gets the current state (playing, paused, or stopped) of the SoundEffectInstance .
	Volume	Gets or sets the volume of the SoundEffectInstance .

See Also

Reference

[SoundEffectInstance Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

SoundEffectInstance.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.IsLooped Property

Gets a value that indicates whether looping is enabled for the [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsLooped { get; set; }
```

Property Value

true if the [SoundEffectInstance](#) has looping enabled; otherwise, **false**.

Remarks

If you want to make a sound play continuously until stopped, be sure to set **IsLooped** to **true** before you call [SoundEffect.Play](#).

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

[How To: Loop a Sound](#)

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Pan Property

Gets or sets the panning for the [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Pan { get; set; }
```

Property Value

Panning, ranging from -1.0f (full left) to 1.0f (full right). 0.0f is centered.

See Also

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Pitch Property

Gets or sets the pitch adjustment for the [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Pitch { get; set; }
```

Property Value

Pitch adjustment, ranging from -1.0f (down one octave) to 1.0f (up one octave). 0.0f is unity (normal) pitch.

See Also

Tasks

[How To: Change the Pitch or Volume of a Sound](#)

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.State Property

Gets the current state (playing, paused, or stopped) of the [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SoundState State { get; }
```

Property Value

[SoundState](#) that represents the current state of the [SoundEffectInstance](#).

See Also

Tasks

[Tutorial 3: Making Sounds with XNA Game Studio](#)

Reference

[SoundState Enumeration](#)

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundEffectInstance.Volume Property

Gets or sets the volume of the [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Volume { get; set; }
```

Property Value

Volume, ranging from 0.0f (silence) to 1.0f (full volume). 1.0f is full volume relative to [SoundEffect.MasterVolume](#).

See Also

Tasks

[How To: Change the Pitch or Volume of a Sound](#)

Reference

[SoundEffect.MasterVolume Property](#)

[SoundEffectInstance Class](#)

[SoundEffectInstance Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SoundState Enumeration

Current state (playing, paused, or stopped) of a [SoundEffectInstance](#).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum SoundState
```

Members

Member name	Description
Paused	The SoundEffectInstance is paused.
Playing	The SoundEffectInstance is playing.
Stopped	The SoundEffectInstance is stopped.

See Also

Reference

[SoundEffectInstance Class](#)

[SoundEffectInstance.Play Method](#)

[SoundEffectInstance.Pause Method](#)

[SoundEffectInstance.Stop Method](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

WaveBank Class

Represents a wave bank, which is a collection of wave files.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class WaveBank : IDisposable
```

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[WaveBank Members](#)


[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista




WaveBank Members

The following tables list the members exposed by the WaveBank type.







Public Constructors

Name	Description
 WaveBank	Overloaded. Initializes a new instance of this class.



Public Properties

Name	Description
 IsDisposed	Gets a value that indicates whether the object is disposed.
 IsInUse	Returns whether the wave bank is currently in use.
 IsPrepared	Returns whether the wave bank is prepared to play.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Tasks

[How To: Play a Sound Using XACT](#)

[How To: Stop or Pause a Sound Using XACT](#)

Concepts

[Audio Overview](#)

Reference

[WaveBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

WaveBank Constructor

Initializes a new instance of this class.

Overload List

Name	Description
WaveBank (AudioEngine, String)	Initializes a new, in-memory instance of this class using a specified AudioEngine and path to a wave bank file.
WaveBank (AudioEngine, String, Int32, Int16)	Initializes a new, streaming instance of this class, using a provided AudioEngine and streaming wave bank parameters.

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

WaveBank Constructor (AudioEngine, String)

Initializes a new, in-memory instance of this class using a specified [AudioEngine](#) and path to a wave bank file.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public WaveBank (
    AudioEngine audioEngine,
    string nonStreamingWaveBankFilename
)
```

Parameters

audioEngine

Instance of an [AudioEngine](#) to associate this wave bank with.

nonStreamingWaveBankFilename

Path to the wave bank file to load.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>audioEngine</i> or <i>nonStreamingWaveBankFilename</i> parameter is null .
InvalidOperationException	Unable to create the WaveBank resource.

Remarks

This constructor generates an in-memory version of a wave bank. The entire wave bank contents are held in memory.

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank Constructor (AudioEngine, String, Int32, Int16)

Initializes a new, streaming instance of this class, using a provided [AudioEngine](#) and streaming wave bank parameters.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public WaveBank (
    AudioEngine audioEngine,
    string streamingWaveBankFilename,
    int offset,
    short packetSize
)
```

Parameters

audioEngine

Instance of an [AudioEngine](#) to associate this wave bank with.

streamingWaveBankFilename

Path to the wave bank file to stream from.

offset

Offset within the wave bank data file. This offset must be DVD sector aligned.

packetSize

Stream packet size, in sectors, to use for each stream. The minimum value is 2.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>audioEngine</i> or <i>nonStreamingWaveBankFilename</i> parameter is null .
InvalidOperationException	Unable to create the WaveBank resource.

Remarks

This constructor constructs a streaming wave bank whose contents are streamed from storage as needed.

When setting *packetSize*, note that the size of a DVD sector is 2,048 bytes. Therefore, setting this value to 2 would result in a packet size of 4,096 bytes. Setting it to 3 would specify packets of 6,144 bytes, setting it to 4 would specify packets of 8,192 bytes, and so on. The optimal DVD size is a multiple of 16 (1 DVD block = 16 DVD sectors).

After creating a streaming wave bank, you must call [Update](#) at least once from the [AudioEngine](#) that was used to create the streaming wave bank before attempting to play a wave from the wave bank. This properly prepares the wave bank for use. Attempts to play waves out of any wave bank before the wave bank is prepared will result in an error.

See Also

Reference

[WaveBank Class](#)







[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[WaveBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

WaveBank.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
WaveBank.Dispose ()	Immediately releases the unmanaged resources used by this object.
WaveBank.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

WaveBank.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[WaveBank Class](#)




[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	IsInUse	Returns whether the wave bank is currently in use.
	IsPrepared	Returns whether the wave bank is prepared to play.

See Also

Reference

[WaveBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

WaveBank.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank.IsInUse Property

Returns whether the wave bank is currently in use.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsInUse { get; }
```

Property Value

true if the wave bank is in use; **false** otherwise.

Remarks

This value will return **true** if the wave bank is currently referenced by at least one valid cue instance or other client. For example, the game itself might reference the wave bank.

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank.IsPrepared Property

Returns whether the wave bank is prepared to play.

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPrepared { get; }
```

Property Value

true if the wave bank is prepared; **false** otherwise.

See Also

Reference

[WaveBank Class](#)


[WaveBank Members](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

WaveBank Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[WaveBank Class](#)

[Microsoft.Xna.Framework.Audio Namespace](#)

WaveBank.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Audio

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[WaveBank Class](#)

[WaveBank Members](#)













[Microsoft.Xna.Framework.Audio Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Microsoft.Xna.Framework.Content Namespace

Contains the run-time components of the Content Pipeline.

Classes

Name	Description
 ContentLoadException	Exception used to report errors from the ContentManager.Load method.
 ContentManager	The ContentManager is the run-time component which loads managed objects from the binary files produced by the design time content pipeline. It also manages the lifespan of the loaded objects, disposing the content manager will also dispose any assets which are themselves IDisposable .
 ContentReader	A worker object that implements most of ContentManager.Load . A new ContentReader is constructed for each asset loaded.
 ContentSerializerAttribute	A custom Attribute that marks a field or property to control how it is serialized or to indicate that protected or private data should be included in serialization.
 ContentSerializerCollectionItemNameAttribute	A custom Attribute that marks a collection class to specify the XML element name for each item in the collection.
 ContentSerializerIgnoreAttribute	A custom Attribute that marks public fields or properties to prevent them from being serialized.
 ContentSerializerRuntimeTypeAttribute	A custom Attribute that specifies the corresponding run-time type of this object.
 ContentSerializerTypeVersionAttribute	A custom Attribute that specifies the corresponding run-time type version of this object.
 ContentTypeReader	Worker for reading a specific managed type from a binary format. Derive from this class to add new data types to the content pipeline system.
 ContentTypeReader	Worker for reading a specific managed type from a binary format.
 ContentTypeReaderManager	A manager that constructs and keeps track of type reader objects.
 ResourceContentManager	Subclass of ContentManager , which is specialized to read from .resx resource files rather than directly from individual files on disk.

See Also

Tasks

[Adding Game Assets to Your Game](#)

[How To: Render a Model](#)

[Using a Custom Importer or Content Processor](#)

[How To: Write a Custom Importer and Processor](#)

Concepts

[Content Pipeline](#)

[Overview of the Content Pipeline](#)

ContentLoadException Class

Exception used to report errors from the [ContentManager.Load](#) method.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]
public class ContentLoadException : Exception
```

See Also

Reference

[ContentLoadException Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








ContentLoadException Members

The following tables list the members exposed by the ContentLoadException type.


Public Constructors

Name	Description
 ContentLoadException	Overloaded. Creates a new instance of ContentLoadException .







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentLoadException Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentLoadException Constructor

Creates a new instance of [ContentLoadException](#).

Overload List

Name	Description
ContentLoadException ()	Creates a new instance of ContentLoadException .
ContentLoadException (SerializationInfo, StreamingContext)	Creates a new instance of ContentLoadException .
ContentLoadException (String)	Creates a new instance of ContentLoadException .
ContentLoadException (String, Exception)	Creates a new instance of ContentLoadException .

See Also

Reference

[ContentLoadException Class](#)

[ContentLoadException Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentLoadException Constructor ()

Creates a new instance of [ContentLoadException](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentLoadException ()
```

See Also

Reference

[ContentLoadException Class](#)

[ContentLoadException Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentLoadException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Creates a new instance of [ContentLoadException](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected ContentLoadException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Describes the value types that were being loaded when the exception occurred.

context

Describes the stream where the exception occurred.

See Also

Reference

[ContentLoadException Class](#)

[ContentLoadException Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentLoadException Constructor (String)

Creates a new instance of [ContentLoadException](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentLoadException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[ContentLoadException Class](#)

[ContentLoadException Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentLoadException Constructor (String, Exception)

Creates a new instance of [ContentLoadException](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentLoadException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The exception that is the cause of the current exception. If the *innerException* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[ContentLoadException Class](#)







[ContentLoadException Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentLoadException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[ContentLoadException Class](#)


[Microsoft.Xna.Framework.Content Namespace](#)

ContentLoadException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[ContentLoadException Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentManager Class

The [ContentManager](#) is the run-time component which loads managed objects from the binary files produced by the design time content pipeline. It also manages the lifespan of the loaded objects, disposing the content manager will also dispose any assets which are themselves [IDisposable](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class ContentManager : IDisposable
```

See Also

Reference

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



ContentManager Members

The following tables list the members exposed by the ContentManager type.









Public Constructors

Name	Description
 ContentManager	Overloaded. Initializes a new instance of ContentManager .





Public Properties

Name	Description
 RootDirectory	Gets or sets the root directory associated with this ContentManager .
 ServiceProvider	Gets the service provider associated with the ContentManager .

Public Methods

Name	Description
 Dispose	Overloaded. Releases the resources used by the ContentManager class.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Load	Loads an asset that has been processed by the Content Pipeline.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Unload	Disposes all data that was loaded by this ContentManager .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 OpenStream	Opens a stream for reading the specified asset. Derived classes can replace this to implement pack files or asset compression.
 ReadAsset	Low-level worker method that reads asset data.

See Also

Reference

[ContentManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentManager Constructor

Initializes a new instance of [ContentManager](#).

Overload List

Name	Description
ContentManager (IServiceProvider)	Initializes a new instance of ContentManager .
ContentManager (IServiceProvider, String)	Initializes a new instance of ContentManager .

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentManager Constructor (IServiceProvider)

Initializes a new instance of [ContentManager](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentManager (
    IServiceProvider serviceProvider
)
```

Parameters

serviceProvider

The service provider that the [ContentManager](#) should use to locate services.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>serviceProvider</i> parameter is null .

Remarks

⚠ Caution

When creating a new [ContentManager](#), if no instance of [Game](#) is otherwise required by the application, it is often better to create a new class that implements the [IServiceProvider](#) interface rather than creating an instance of [Game](#) just to create a new instance of [GraphicsDeviceManager](#).

By default, the [ContentManager](#) searches for content in the directory where the executable is located.

Example

To create a [ContentManager](#) that uses [Game.Services](#):

C#

```
ContentManager contentManager = new ContentManager( Services );
```

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager Constructor (IServiceProvider, String)

Initializes a new instance of [ContentManager](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentManager (
    IServiceProvider serviceProvider,
    string rootDirectory
)
```

Parameters

serviceProvider

The service provider the [ContentManager](#) should use to locate services.

rootDirectory

The root directory to search for content.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>serviceProvider</i> or <i>rootDirectory</i> parameter is null .

Remarks

⚠ Caution

When creating a new [ContentManager](#), if no instance of [Game](#) is otherwise required by the application, it is often better to create a new class that implements the [IServiceProvider](#) interface rather than creating an instance of [Game](#) just to create a new instance of [GraphicsDeviceManager](#).

Example

To create a [ContentManager](#) that uses [Game.Services](#) and searches for resources from the root of the C: drive:

C#

```
ContentManager contentManager = new ContentManager( Services, "C:\\\" );
```

See Also

Reference

[ContentManager Class](#)









[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager Methods

Public Methods

Name	Description
 Dispose	Overloaded. Releases the resources used by the ContentManager class.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Load	Loads an asset that has been processed by the Content Pipeline.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Unload	Disposes all data that was loaded by this ContentManager .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 OpenStream	Opens a stream for reading the specified asset. Derived classes can replace this to implement pack files or asset compression.
 ReadAsset	Low-level worker method that reads asset data.

See Also

Reference

[ContentManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentManager.Dispose Method

Releases the resources used by the [ContentManager](#) class.

Overload List

Name	Description
ContentManager.Dispose ()	Releases all resources used by the ContentManager class.
ContentManager.Dispose (Boolean)	Releases the unmanaged resources used by the ContentManager and optionally releases the managed resources.

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentManager.Dispose Method ()

Releases all resources used by the [ContentManager](#) class.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager.Dispose Method (Boolean)

Releases the unmanaged resources used by the [ContentManager](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager.Load Generic Method

Loads an asset that has been processed by the Content Pipeline.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual T Load<T> (
    string assetName
)
```

Type Parameters

T

The type of asset to load. [Model](#), [Effect](#), [SpriteFont](#), [Texture](#), [Texture2D](#), [Texture3D](#) and [TextureCube](#) are all supported by default by the standard Content Pipeline processor, but additional types may be loaded by extending the processor.

Parameters

assetName

Asset name, relative to the loader root directory, and not including the .xnb file extension.

Return Value

The loaded asset. Repeated calls to load the same asset will return the same object instance.


Exceptions

Exception type	Condition
ObjectDisposedException	Load was called after the ContentManager was disposed.
ArgumentNullException	The <i>assetName</i> argument is null .
ContentLoadException	The type of the <i>assetName</i> in the file does not match the type of asset requested as specified by <i>T</i> .

Remarks

Before a [ContentManager](#) can load an asset, you need to add the asset to your game project using the steps described in [Adding Game Assets to Your Game](#).

The following are the Content Pipeline run-time classes supported by **Load** and the file formats they are associated with.

Effect Class	.fx
Model Class	.fbx, .x
SpriteFont Class	.bmp, .spritefont, .dds, .dib, .hdr, .jpg, .pfm, .png, .ppm, and .tga  Note The default content processor for .bmp files is the texture processor. To process a .bmp file for use as a Sprite Font, click Sprite Font Texture - XNA Framework in the Properties pane for the .bmp file after it has been added to the project.
Texture Class, Texture2D Class	.bmp, .dds, .dib, .hdr, .jpg, .pfm, .png, .ppm, and .tga
Texture3D Class, TextureCube Class	.dds

Note that these are the formats of the original assets; after processing, all assets will be .xnb files.

Example

To load a [Model](#) from the directory .\content\models\ with the asset name of "box":

C#

```
Model model = contentManager.Load<Model>( ".\\content\\models\\box" );
```

See Also

Tasks

[Adding Game Assets to Your Game](#)

[How To: Write a Custom Importer and Processor](#)

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager.OpenStream Method

Opens a stream for reading the specified asset. Derived classes can replace this to implement pack files or asset compression.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual Stream OpenStream (  
    string assetName  
)
```

Parameters

assetName

The name of the asset being read.

Return Value

The opened stream.

Exceptions

Exception type	Condition
ContentLoadException	Unable to open the file containing <i>assetName</i> , or the file was not found.

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager.ReadAsset Generic Method

Low-level worker method that reads asset data.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected T ReadAsset<T> (
    string assetName,
    Action<IDisposable> recordDisposableObject
)
```

Type Parameters

T

Type of the asset specified for loading.

Parameters

assetName

The name of the asset to be loaded from disk.

recordDisposableObject

Delegate function for handling the disposition of assets. If *recordDisposableObject* is **null**, the default lifespan tracking and management is used, so unloading or disposing of the content manager frees everything that has been loaded through it. If *recordDisposableObject* specifies a valid delegate, that delegate is used instead of the default lifespan tracking and is called every time the loader encounters a type that implements [IDisposable](#). You must use your own code to unload assets loaded in this fashion, since [ContentManager's Unload](#) method will not be aware of them.

Return Value

Returns the loaded asset.

Exceptions

Exception type	Condition
ObjectDisposedException	ReadAsset was called after the ContentManager was disposed.
ArgumentNullException	The <i>assetName</i> argument is null .

Remarks

This protected method loads an .xnb file from disk. Unlike [ContentManager's Load](#) method, it always returns a new copy of the specified asset, even if the asset has been loaded previously.

You can customize how shared assets are managed by overriding [ContentManager's Load](#) method. In your own `Load` method, keep track of what has been loaded, and call **ReadAsset** when you need a new asset from disk.

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager.Unload Method

Disposes all data that was loaded by this [ContentManager](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual void Unload ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	Unload was called after the ContentManager was disposed.

See Also

Reference

[ContentManager Class](#)



[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager Properties

Public Properties

	Name	Description
	RootDirectory	Gets or sets the root directory associated with this ContentManager .
	ServiceProvider	Gets the service provider associated with the ContentManager .

See Also

Reference

[ContentManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentManager.RootDirectory Property

Gets or sets the root directory associated with this [ContentManager](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string RootDirectory { get; set; }
```

Property Value

The root directory associated with this [ContentManager](#).

Exceptions

Exception type	Condition
ArgumentNullException	RootDirectory cannot be null .
InvalidOperationException	This property cannot be changed after content has been loaded into the ContentManager .

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentManager.ServiceProvider Property

Gets the service provider associated with the [ContentManager](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IServiceProvider ServiceProvider { get; }
```

Property Value

The service provider associated with the [ContentManager](#).

See Also

Reference

[ContentManager Class](#)

[ContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader Class

A worker object that implements most of [ContentManager.Load](#). A new [ContentReader](#) is constructed for each asset loaded.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ContentReader : BinaryReader
```

See Also

Reference

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide




[Content Pipeline](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune


































ContentReader Members

The following tables list the members exposed by the ContentReader type.

Public Properties





Name	Description
 AssetName	Gets the name of the asset currently being read by this ContentReader .
 BaseStream	(Inherited from BinaryReader .)
 ContentManager	Gets the ContentManager associated with the ContentReader .

Public Methods

Name	Description
 Close	(Inherited from BinaryReader .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 PeekChar	(Inherited from BinaryReader .)
 Read	(Inherited from BinaryReader .)
 ReadBoolean	(Inherited from BinaryReader .)
 ReadByte	(Inherited from BinaryReader .)
 ReadBytes	(Inherited from BinaryReader .)
 ReadChar	(Inherited from BinaryReader .)
 ReadChars	(Inherited from BinaryReader .)
 ReadColor	Reads a Color value from the currently open stream.
 ReadDecimal	(Inherited from BinaryReader .)
 ReadDouble	Reads a double value from the currently open stream.
 ReadExternalReference	Reads a link to an external file.
 ReadInt16	(Inherited from BinaryReader .)
 ReadInt32	(Inherited from BinaryReader .)
 ReadInt64	(Inherited from BinaryReader .)
 ReadMatrix	Reads a Matrix value from the currently open stream.
 ReadObject	Overloaded. Reads a single managed object from the current stream. Can be called recursively.
 ReadQuaternion	Reads a Quaternion value from the current stream.
 ReadRawObject	Overloaded. Reads a single managed object from the current stream as an instance of the specified type. If a base class of the actual object type is specified only data from the base type will be read.
 ReadSByte	(Inherited from BinaryReader .)
 ReadSharedResource	Reads a shared resource ID, and records it for subsequent fix-up.
 ReadSingle	Reads a float value from the currently open stream.
 ReadString	(Inherited from BinaryReader .)
 ReadUInt16	(Inherited from BinaryReader .)
 ReadUInt32	(Inherited from BinaryReader .)
 ReadUInt64	(Inherited from BinaryReader .)
 ReadVector2	Reads a Vector2 value from the current stream.
 ReadVector3	Reads a Vector3 value from the current stream.
 ReadVector4	Reads a Vector4 value from the current stream.
 ReferenceEquals	(Inherited from Object .)

 [ToString](#) (Inherited from [Object](#).)

Protected Methods

Name	Description
 Dispose	(Inherited from BinaryReader .)
 FillBuffer	(Inherited from BinaryReader .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also



































Reference

[ContentReader Class](#)





[Microsoft.Xna.Framework.Content Namespace](#)

ContentReader Methods

Public Methods

Name	Description
 Close	(Inherited from BinaryReader .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 PeekChar	(Inherited from BinaryReader .)
 Read	(Inherited from BinaryReader .)
 ReadBoolean	(Inherited from BinaryReader .)
 ReadByte	(Inherited from BinaryReader .)
 ReadBytes	(Inherited from BinaryReader .)
 ReadChar	(Inherited from BinaryReader .)
 ReadChars	(Inherited from BinaryReader .)
 ReadColor	Reads a Color value from the currently open stream.
 ReadDecimal	(Inherited from BinaryReader .)
 ReadDouble	Reads a double value from the currently open stream.
 ReadExternalReference	Reads a link to an external file.
 ReadInt16	(Inherited from BinaryReader .)
 ReadInt32	(Inherited from BinaryReader .)
 ReadInt64	(Inherited from BinaryReader .)
 ReadMatrix	Reads a Matrix value from the currently open stream.
 ReadObject	Overloaded. Reads a single managed object from the current stream. Can be called recursively.
 ReadQuaternion	Reads a Quaternion value from the current stream.
 ReadRawObject	Overloaded. Reads a single managed object from the current stream as an instance of the specified type. If a base class of the actual object type is specified only data from the base type will be read.
 ReadSByte	(Inherited from BinaryReader .)
 ReadSharedResource	Reads a shared resource ID, and records it for subsequent fix-up.
 ReadSingle	Reads a float value from the currently open stream.
 ReadString	(Inherited from BinaryReader .)
 ReadUInt16	(Inherited from BinaryReader .)
 ReadUInt32	(Inherited from BinaryReader .)
 ReadUInt64	(Inherited from BinaryReader .)
 ReadVector2	Reads a Vector2 value from the current stream.
 ReadVector3	Reads a Vector3 value from the current stream.
 ReadVector4	Reads a Vector4 value from the current stream.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Dispose	(Inherited from BinaryReader .)
 FillBuffer	(Inherited from BinaryReader .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentReader Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentReader.ReadColor Method

Reads a [Color](#) value from the currently open stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color ReadColor ()
```

Return Value

The [Color](#) value that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadDouble Method

Reads a **double** value from the currently open stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override double ReadDouble ()
```

Return Value

The **double** value that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadExternalReference Generic Method

Reads a link to an external file.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadExternalReference<T> ()
```

Type Parameters

T

The type of asset stored in the external file.

Return Value

The asset stored in the external file.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadMatrix Method

Reads a [Matrix](#) value from the currently open stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix ReadMatrix ()
```

Return Value

The [Matrix](#) that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadObject Method

Reads a single managed object from the current stream. Can be called recursively.

Overload List

Name	Description
ContentReader.ReadObject ()	Reads a single managed object from the current stream. Can be called recursively.
ContentReader.ReadObject (ContentTypeReader)	Reads a single managed object from the current stream. Can be called recursively.
ContentReader.ReadObject (ContentTypeReader, T)	Reads a single managed object from the current stream. Can be called recursively.
ContentReader.ReadObject (T)	Reads a single managed object from the current stream. Can be called recursively.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentReader.ReadObject Generic Method ()

Reads a single managed object from the current stream. Can be called recursively.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadObject<T> ()
```

Type Parameters

T

The type of object to read.

Return Value

The object that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadObject Generic Method (ContentTypeReader)

Reads a single managed object from the current stream. Can be called recursively.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadObject<T> (  
    ContentTypeReader typeReader  
)
```

Type Parameters

T

The type of object to read.

Parameters

typeReader

The [ContentTypeReader](#) to use to read the object.

Return Value

The object that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadObject Generic Method (ContentTypeReader, T)

Reads a single managed object from the current stream. Can be called recursively.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadObject<T> (  
    ContentTypeReader typeReader,  
    T existingInstance  
)
```

Type Parameters

T

The type of object to read.

Parameters

typeReader

The [ContentTypeReader](#) to use to read the object.

existingInstance

An existing object to write into.

Return Value

The object that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadObject Generic Method (T)

Reads a single managed object from the current stream. Can be called recursively.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadObject<T> (  
    T existingInstance  
)
```

Type Parameters

T

The type of object to read.

Parameters

existingInstance

An existing object to write into.

Return Value

The object that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadQuaternion Method

Reads a [Quaternion](#) value from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion ReadQuaternion ()
```

Return Value

The [Quaternion](#) that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadRawObject Method

Reads a single managed object from the current stream as an instance of the specified type. If a base class of the actual object type is specified only data from the base type will be read.

Overload List

Name	Description
ContentReader.ReadRawObject ()	Reads a single managed object from the current stream as an instance of the specified type. If you specify a base class of the actual object type, this method reads data only from the base type.
ContentReader.ReadRawObject (ContentTypeReader)	Reads a single managed object from the current stream as an instance of the specified type. If a base class of the actual object type is specified only data from the base type will be read.
ContentReader.ReadRawObject (ContentTypeReader, T)	Reads a single managed object from the current stream as an instance of the specified type. If you specify a base class of the actual object type, this method reads data only from the base type.
ContentReader.ReadRawObject (T)	Reads a single managed object from the current stream as an instance of the specified type. If you specify a base class of the actual object type, the method reads data only from the base type.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentReader.ReadRawObject Generic Method ()

Reads a single managed object from the current stream as an instance of the specified type. If you specify a base class of the actual object type, this method reads data only from the base type.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadRawObject<T> ()
```

Type Parameters

T

The type of object to read.

Return Value

The object that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadRawObject Generic Method (ContentTypeReader)

Reads a single managed object from the current stream as an instance of the specified type. If a base class of the actual object type is specified only data from the base type will be read.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    ContentTypeReader typeReader  
)
```

Type Parameters

T

The type of object to read.

Parameters

typeReader

The [ContentTypeReader](#) to use to read the object.

Return Value

The object that was read.

Exceptions

Exception type	Condition
ArgumentNullException	<i>typeReader</i> is null .

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadRawObject Generic Method (ContentTypeReader, T)

Reads a single managed object from the current stream as an instance of the specified type. If you specify a base class of the actual object type, this method reads data only from the base type.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    ContentTypeReader typeReader,  
    T existingInstance  
)
```

Type Parameters

T

The type of object to read.

Parameters

typeReader

The [ContentTypeReader](#) to use to read the object.

existingInstance

An existing object to write into.

Return Value

The object that was read.

Exceptions

Exception type	Condition
ArgumentNullException	<i>typeReader</i> is null .

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadRawObject Generic Method (T)

Reads a single managed object from the current stream as an instance of the specified type. If you specify a base class of the actual object type, the method reads data only from the base type.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    T existingInstance  
)
```

Type Parameters

T

The type of object to read.

Parameters

existingInstance

An existing object to write into.

Return Value

The object that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadSharedResource Generic Method

Reads a shared resource ID, and records it for subsequent fix-up.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ReadSharedResource<T> (  
    Action<T> fixup  
)
```

Type Parameters

T

The type of the shared resource.

Parameters

fixup

The fix-up action to perform.

Exceptions

Exception type	Condition
ArgumentNullException	<i>fixup</i> is null .
ContentLoadException	Error loading the resource.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadSingle Method

Reads a **float** value from the currently open stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override float ReadSingle ()
```

Return Value

The **float** value that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadVector2 Method

Reads a [Vector2](#) value from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ReadVector2 ()
```

Return Value

The [Vector2](#) that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadVector3 Method

Reads a [Vector3](#) value from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 ReadVector3 ()
```

Return Value

The [Vector3](#) that was read.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ReadVector4 Method

Reads a [Vector4](#) value from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ReadVector4 ()
```

Return Value

The [Vector4](#) that was read.

See Also

Reference

[ContentReader Class](#)




[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader Properties

Public Properties

	Name	Description
	AssetName	Gets the name of the asset currently being read by this ContentReader .
	BaseStream	(Inherited from BinaryReader .)
	ContentManager	Gets the ContentManager associated with the ContentReader .

See Also

Reference

[ContentReader Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentReader.AssetName Property

Gets the name of the asset currently being read by this [ContentReader](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string AssetName { get; }
```

Property Value

Name of the asset currently being read by this [ContentReader](#).

Remarks

AssetName can be useful to include in the exception message if a custom [ContentTypeReader](#) encounters an error.

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentReader.ContentManager Property

Gets the [ContentManager](#) associated with the [ContentReader](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentManager ContentManager { get; }
```

Property Value

The [ContentManager](#) associated with the [ContentReader](#).

See Also

Reference

[ContentReader Class](#)

[ContentReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute Class

A custom [Attribute](#) that marks a field or property to control how it is serialized or to indicate that protected or private data should be included in serialization.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[AttributeUsageAttribute(384)]  
public sealed class ContentSerializerAttribute : Attribute
```

See Also

Reference

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









ContentSerializerAttribute Members

The following tables list the members exposed by the ContentSerializerAttribute type.











Public Constructors

Name	Description
 ContentSerializerAttribute	Creates a new instance of ContentSerializerAttribute .



Public Properties

Name	Description
 AllowNull	Get or set a value indicating whether this member can have a null value (default=true).
 CollectionItemName	Gets or sets the XML element name for each item in a collection (default = "Item").
 ElementName	Gets or sets the XML element name (default=name of the managed type member).
 FlattenContent	Gets or sets a value indicating whether to write member contents directly into the current XML context rather than wrapping the member in a new XML element (default=false).
 HasCollectionItemName	Indicates whether an explicit CollectionItemName string is being used or the default value.
 Optional	Indicates whether to write this element if the member is null and skip past it if not found when deserializing XML (default=false).
 SharedResource	Indicates whether this member is referenced from multiple parents and should be serialized as a unique ID reference (default=false).
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Clone	Creates a copy of the ContentSerializerAttribute .
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentSerializerAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerAttribute Constructor

Creates a new instance of [ContentSerializerAttribute](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentSerializerAttribute ()
```

Example

C#

```
[ContentSerializerAttribute()]
```

See Also

Reference

[ContentSerializerAttribute Class](#)











[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute Methods

Public Methods

	Name	Description
	Clone	Creates a copy of the ContentSerializerAttribute .
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentSerializerAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerAttribute.Clone Method

Creates a copy of the [ContentSerializerAttribute](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentSerializerAttribute Clone ()
```

Return Value

The copy.

See Also

Reference

[ContentSerializerAttribute Class](#)









[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute Properties

Public Properties

	Name	Description
	AllowNull	Get or set a value indicating whether this member can have a null value (default=true).
	CollectionItemName	Gets or sets the XML element name for each item in a collection (default = "Item").
	ElementName	Gets or sets the XML element name (default=name of the managed type member).
	FlattenContent	Gets or sets a value indicating whether to write member contents directly into the current XML context rather than wrapping the member in a new XML element (default=false).
	HasCollectionItemName	Indicates whether an explicit CollectionItemName string is being used or the default value.
	Optional	Indicates whether to write this element if the member is null and skip past it if not found when deserializing XML (default=false).
	SharedResource	Indicates whether this member is referenced from multiple parents and should be serialized as a unique ID reference (default=false).
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentSerializerAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerAttribute.AllowNull Property

Get or set a value indicating whether this member can have a null value (default=true).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AllowNull { get; set; }
```

Property Value

Value indicating whether this member can have a null value.

Example

[ContentSerializerAttribute](#) with [AllowNull](#) set to **false**.

C#

```
[ContentSerializerAttribute( AllowNull = false )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute.CollectionItemName Property

Gets or sets the XML element name for each item in a collection (default = "Item").

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string CollectionItemName { get; set; }
```

Property Value

The XML element name for each item in the collection.

Exceptions

Exception type	Condition
ArgumentNullException	<i>CollectionItemName</i> is null .

Example

[ContentSerializerAttribute](#) with [CollectionItemName](#) set to the string "PlayerName":

C#

```
[ContentSerializerAttribute( CollectionItemName = "PlayerName" )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute.ElementName Property

Gets or sets the XML element name (default=name of the managed type member).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string ElementName { get; set; }
```

Property Value

The XML element name.

Example

[ContentSerializerAttribute](#) with [ElementName](#) set to the string "PlayerName".

C#

```
[ContentSerializerAttribute( ElementName = "PlayerName" )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute.FlattenContent Property

Gets or sets a value indicating whether to write member contents directly into the current XML context rather than wrapping the member in a new XML element (default=false).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool FlattenContent { get; set; }
```

Property Value

Value indicating whether to write member contents directly into the current XML context rather than wrapping the member in a new XML element.

Example

[ContentSerializerAttribute](#) with [FlattenContent](#) set **true**.

C#

```
[ContentSerializerAttribute( FlattenContent = true )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute.HasCollectionItemName Property

Indicates whether an explicit CollectionItemName string is being used or the default value.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasCollectionItemName { get; }
```

Property Value

true if an explicit CollectionItemName string is being used; **false** otherwise.

Example

[ContentSerializerAttribute](#) with [HasCollectionItemName](#) set **true**.

C#

```
[ContentSerializerAttribute( HasCollectionItemName = true )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute.Optional Property

Indicates whether to write this element if the member is **null** and skip past it if not found when deserializing XML (default=false).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Optional { get; set; }
```

Property Value

true if the member should be skipped when **null**.

Example

[ContentSerializerAttribute](#) with [Optional](#) set **true**.

C#

```
[ContentSerializerAttribute( Optional = true )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerAttribute.SharedResource Property

Indicates whether this member is referenced from multiple parents and should be serialized as a unique ID reference (default=false).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SharedResource { get; set; }
```

Property Value

true if this member is referenced from multiple parents; **false** otherwise.

Example

[ContentSerializerAttribute](#) with [SharedResource](#) set **true**.

C#

```
[ContentSerializerAttribute( SharedResource = true )]
```

See Also

Reference

[ContentSerializerAttribute Class](#)

[ContentSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerCollectionItemNameAttribute Class

A custom [Attribute](#) that marks a collection class to specify the XML element name for each item in the collection.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[AttributeUsageAttribute(4)]  
public sealed class ContentSerializerCollectionItemNameAttribute : Attribute
```

See Also

Reference

[ContentSerializerCollectionItemNameAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



ContentSerializerCollectionItemNameAttribute Members

The following tables list the members exposed by the ContentSerializerCollectionItemNameAttribute type.










Public Constructors

Name	Description
 ContentSerializerCollectionItemNameAttribute	Creates a new instance of ContentSerializerCollectionItemNameAttribute .



Public Properties

Name	Description
 CollectionItemName	Gets the name that will be used for each item in the collection.
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentSerializerCollectionItemNameAttribute Class](#)
[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerCollectionItemNameAttribute Constructor

Creates a new instance of [ContentSerializerCollectionItemNameAttribute](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentSerializerCollectionItemNameAttribute (
    string collectionItemName
)
```

Parameters

collectionItemName

The name for each item in the collection.

Exceptions

Exception type	Condition
ArgumentNullException	<i>collectionItemName</i> is null .

Example

C#

```
[ContentSerializerCollectionItemNameAttribute( "CollectionItem" )]
```

See Also

Reference

[ContentSerializerCollectionItemNameAttribute Class](#)










[ContentSerializerCollectionItemNameAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerCollectionItemNameAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[ContentSerializerCollectionItemNameAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerCollectionItemNameAttribute Properties

Public Properties

	Name	Description
	CollectionItemName	Gets the name that will be used for each item in the collection.
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentSerializerCollectionItemNameAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerCollectionItemNameAttribute.CollectionItemName Property

Gets the name that will be used for each item in the collection.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string CollectionItemName { get; }
```

Property Value

The name used for each item in the collection.

See Also

Reference

[ContentSerializerCollectionItemNameAttribute Class](#)

[ContentSerializerCollectionItemNameAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerIgnoreAttribute Class

A custom [Attribute](#) that marks public fields or properties to prevent them from being serialized.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[AttributeUsageAttribute(384)]  
public sealed class ContentSerializerIgnoreAttribute : Attribute
```

See Also

Reference

[ContentSerializerIgnoreAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerIgnoreAttribute Members

The following tables list the members exposed by the ContentSerializerIgnoreAttribute type.










Public Constructors

Name	Description
 ContentSerializerIgnoreAttribute	Creates a new instance of ContentSerializerIgnoreAttribute .



Public Properties

Name	Description
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentSerializerIgnoreAttribute Class](#)
[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerIgnoreAttribute Constructor

Creates a new instance of [ContentSerializerIgnoreAttribute](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentSerializerIgnoreAttribute ()
```

Example

C#

```
[ContentSerializerIgnoreAttribute()]
```

See Also

Reference

[ContentSerializerIgnoreAttribute Class](#)










[ContentSerializerIgnoreAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerIgnoreAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[ContentSerializerIgnoreAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerIgnoreAttribute Properties

Public Properties

	Name	Description
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentSerializerIgnoreAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerRuntimeTypeAttribute Class

A custom [Attribute](#) that specifies the corresponding run-time type of this object. For more information, see [Automatic Serialization of .XNB Files](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ContentSerializerRuntimeTypeAttribute : Attribute
```

Remarks

You can apply multiple instances of this attribute to a single class, thus specifying different strong names for each target platform. If you don't specify the target platform, a single attribute provides a strong name for all platforms.

See Also

Reference

[ContentSerializerRuntimeTypeAttribute Members](#)


[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



ContentSerializerRuntimeTypeAttribute Members

The following tables list the members exposed by the ContentSerializerRuntimeTypeAttribute type.










Public Constructors

Name	Description
 ContentSerializerRuntimeTypeAttribute	Creates a new instance of ContentSerializerRuntimeTypeAttribute.



Public Properties

Name	Description
 RuntimeType	Gets the run-time type for the object.
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentSerializerRuntimeTypeAttribute Class](#)
[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerRuntimeTypeAttribute Constructor

Creates a new instance of **ContentSerializerRuntimeTypeAttribute**.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentSerializerRuntimeTypeAttribute (  
    string runtimeType  
)
```

Parameters

runtimeType

The run-time type name of the object.

See Also

Reference

[ContentSerializerRuntimeTypeAttribute Class](#)










[ContentSerializerRuntimeTypeAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerRuntimeTypeAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[ContentSerializerRuntimeTypeAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerRuntimeTypeAttribute Properties

Public Properties

	Name	Description
	RuntimeType	Gets the run-time type for the object.
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentSerializerRuntimeTypeAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerRuntimeTypeAttribute.RuntimeType Property

Gets the run-time type for the object.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string RuntimeType { get; }
```

Property Value

The run-time type name of the object.

See Also

Reference

[ContentSerializerRuntimeTypeAttribute Class](#)

[ContentSerializerRuntimeTypeAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerTypeVersionAttribute Class

A custom [Attribute](#) that specifies the corresponding run-time type version of this object. For more information, see [Automatic Serialization of .XNB Files](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ContentSerializerTypeVersionAttribute : Attribute
```

See Also

Reference

[ContentSerializerTypeVersionAttribute Members](#)


[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



ContentSerializerTypeVersionAttribute Members

The following tables list the members exposed by the ContentSerializerTypeVersionAttribute type.










Public Constructors

Name	Description
 ContentSerializerTypeVersionAttribute	Creates a new instance of ContentSerializerTypeVersionAttribute.



Public Properties

Name	Description
 TypeId	(Inherited from Attribute .)
 TypeVersion	Gets the run-time type version for the object.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentSerializerTypeVersionAttribute Class](#)
[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerTypeVersionAttribute Constructor

Creates a new instance of **ContentSerializerTypeVersionAttribute**.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentSerializerTypeVersionAttribute (  
    int typeVersion  
)
```

Parameters

typeVersion

The run-time type version of the object.

See Also

Reference

[ContentSerializerTypeVersionAttribute Class](#)










[ContentSerializerTypeVersionAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentSerializerTypeVersionAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[ContentSerializerTypeVersionAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerTypeVersionAttribute Properties

Public Properties

	Name	Description
	TypeId	(Inherited from Attribute .)
	TypeVersion	Gets the run-time type version for the object.

See Also

Reference

[ContentSerializerTypeVersionAttribute Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentSerializerTypeVersionAttribute.TypeVersion Property

Gets the run-time type version for the object.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int TypeVersion { get; }
```

Property Value

The run-time type version of the object.

See Also

Reference

[ContentSerializerTypeVersionAttribute Class](#)

[ContentSerializerTypeVersionAttribute Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader Class

Worker for reading a specific managed type from a binary format.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class ContentTypeReader
```

See Also

Reference

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




ContentTypeReader Members

The following tables list the members exposed by the ContentTypeReader type.






Protected Constructors

Name	Description
 ContentTypeReader	Creates a new instance of ContentTypeReader .





Public Properties

Name	Description
 CanDeserializeIntoExistingObject	Determines if deserialization into an existing object is possible.
 TargetType	Gets the type handled by this reader component.
 TypeVersion	Gets a format version number for this type.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 Initialize	Retrieves and caches nested type readers. Called by the framework at creation time.
 MemberwiseClone	(Inherited from Object .)
 Read	Reads a strongly typed object from the current stream.

See Also

Reference

[ContentTypeReader Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReader Constructor

Creates a new instance of [ContentTypeReader](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected ContentTypeReader (  
    Type targetType  
)
```

Parameters

targetType

The type handled by this reader component.

See Also

Reference

[ContentTypeReader Class](#)






[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	Initialize	Retrieves and caches nested type readers. Called by the framework at creation time.
	MemberwiseClone	(Inherited from Object .)
	Read	Reads a strongly typed object from the current stream.

See Also

Reference

[ContentTypeReader Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReader.Initialize Method

Retrieves and caches nested type readers. Called by the framework at creation time.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected internal virtual void Initialize (  
    ContentTypeReaderManager manager  
)
```

Parameters

manager

The content manager.

See Also

Reference

[ContentTypeReader Class](#)

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader.Read Method

Reads a strongly typed object from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected internal abstract Object Read (  
    ContentReader input,  
    Object existingInstance  
)
```

Parameters

input

The [ContentReader](#) used to read the object.

existingInstance

The object receiving the data, or **null** if a new instance of the object should be created.

Return Value

The object that was read.

See Also

Reference

[ContentTypeReader Class](#)




[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader Properties

Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	Determines if deserialization into an existing object is possible.
	TargetType	Gets the type handled by this reader component.
	TypeVersion	Gets a format version number for this type.

See Also

Reference

[ContentTypeReader Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReader.CanDeserializeIntoExistingObject Property

Determines if deserialization into an existing object is possible.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual bool CanDeserializeIntoExistingObject { get; }
```

Property Value

true if the object can be deserialized into; **false** otherwise.

See Also

Reference

[ContentTypeReader Class](#)

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader.TargetType Property

Gets the type handled by this reader component.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Type TargetType { get; }
```

Property Value

The type handled by this reader component.

See Also

Reference

[ContentTypeReader Class](#)

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader.TypeVersion Property

Gets a format version number for this type.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual int TypeVersion { get; }
```

Property Value

The version number for this type.

See Also

Reference

[ContentTypeReader Class](#)

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader Generic Class

Worker for reading a specific managed type from a binary format. Derive from this class to add new data types to the content pipeline system.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class ContentTypeReader<T> : ContentTypeReader
```

See Also

Reference

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide


[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




ContentTypeReader Members

The following tables list the members exposed by the ContentTypeReader type.







Protected Constructors

	Name	Description
	ContentTypeReader	Creates a new instance of ContentTypeReader .



Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	(Inherited from ContentTypeReader .)
	TargetType	(Inherited from ContentTypeReader .)
	TypeVersion	(Inherited from ContentTypeReader .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Read	Overloaded. Reads a strongly typed object from the current stream.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeReader Generic Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReader Constructor

Creates a new instance of [ContentTypeReader](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected ContentTypeReader ()
```

See Also

Reference

[ContentTypeReader Generic Class](#)







[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Read	Overloaded. Reads a strongly typed object from the current stream.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeReader Generic Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReader.Read Method

Reads a strongly typed object from the current stream.

Overload List

Name	Description
ContentTypeReader.Read (ContentReader, Object)	Reads an object from the current stream.
ContentTypeReader.Read (ContentReader, T)	Reads a strongly typed object from the current stream.

See Also

Reference

[ContentTypeReader Generic Class](#)

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReader.Read Method (ContentReader, Object)

Reads an object from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected internal override Object Read (  
    ContentReader input,  
    Object existingInstance  
)
```

Parameters

input

The [ContentReader](#) used to read the object.

existingInstance

An existing object to read into.

Return Value

The object that was read.

Exceptions

Exception type	Condition
ContentLoadException	Error loading the object. File contains an object of a type other than the type requested.

See Also

Reference

[ContentTypeReader Generic Class](#)

[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader.Read Method (ContentReader, T)

Reads a strongly typed object from the current stream.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected internal abstract T Read (  
    ContentReader input,  
    T existingInstance  
)
```

Parameters

input

The [ContentReader](#) used to read the object.

existingInstance

An existing object to read into.

Return Value

The type of object to read.

See Also

Reference

[ContentTypeReader Generic Class](#)




[ContentTypeReader Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ContentTypeReader Properties

Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	(Inherited from ContentTypeReader .)
	TargetType	(Inherited from ContentTypeReader .)
	TypeVersion	(Inherited from ContentTypeReader .)

See Also

Reference

[ContentTypeReader Generic Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReaderManager Class

A manager that constructs and keeps track of type reader objects.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ContentTypeReaderManager
```

See Also

Reference

[ContentTypeReaderManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide







[Content Pipeline](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



ContentTypeReaderManager Members

The following tables list the members exposed by the ContentTypeReaderManager type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetTypeReader	Looks up a reader for the specified type.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[ContentTypeReaderManager Class](#)



[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReaderManager Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetTypeReader	Looks up a reader for the specified type.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeReaderManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ContentTypeReaderManager.GetTypeReader Method

Looks up a reader for the specified type.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ContentTypeReader GetTypeReader (  
    Type targetType  
)
```

Parameters

targetType

The type the reader will handle.

Return Value

The created [ContentTypeReader](#).

Exceptions

Exception type	Condition
ContentLoadException	Cannot find the requested ContentTypeReader .
ArgumentNullException	<i>targetType</i> cannot be null .

See Also

Reference

[ContentTypeReaderManager Class](#)

[ContentTypeReaderManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceContentManager Class

Subclass of [ContentManager](#), which is specialized to read from .resx resource files rather than directly from individual files on disk.

See [How To: Load Content](#) for instructions on using **ResourceContentManager**.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class ResourceContentManager : ContentManager
```

See Also

Reference

[ResourceContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Programming Guide

[Content Pipeline](#)


[How To: Load Content](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



ResourceContentManager Members

The following tables list the members exposed by the ResourceContentManager type.









Public Constructors

Name	Description
 ResourceContentManager	Creates a new instance of ResourceContentManager .





Public Properties

Name	Description
 RootDirectory	(Inherited from ContentManager .)
 ServiceProvider	(Inherited from ContentManager .)

Public Methods

Name	Description
 Dispose	(Inherited from ContentManager .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Load	(Inherited from ContentManager .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Unload	(Inherited from ContentManager .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 OpenStream	Opens a stream for reading the specified resource. Derived classes can replace this to implement pack files or asset compression.
 ReadAsset	(Inherited from ContentManager .)

See Also

Reference

[ResourceContentManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ResourceContentManager Constructor

Creates a new instance of [ResourceContentManager](#).

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ResourceContentManager (
    IServiceProvider serviceProvider,
    ResourceManager resourceManager
)
```

Parameters

serviceProvider

The service provider the [ContentManager](#) should use to locate services.

resourceManager

The resource manager for the [ResourceContentManager](#) to read from.

Exceptions

Exception type	Condition
ArgumentNullException	<i>resourceManager</i> is null .

See Also

Reference

[ResourceContentManager Class](#)









[ResourceContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceContentManager Methods

Public Methods

	Name	Description
	Dispose	(Inherited from ContentManager .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Load	(Inherited from ContentManager .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Unload	(Inherited from ContentManager .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	OpenStream	Opens a stream for reading the specified resource. Derived classes can replace this to implement pack files or asset compression.
	ReadAsset	(Inherited from ContentManager .)

See Also

Reference

[ResourceContentManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)

ResourceContentManager.OpenStream Method

Opens a stream for reading the specified resource. Derived classes can replace this to implement pack files or asset compression.

Namespace: Microsoft.Xna.Framework.Content

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override Stream OpenStream (  
    string assetName  
)
```

Parameters

assetName

The name of the asset being read.

Return Value

The opened stream.

Exceptions

Exception type	Condition
ContentLoadException	Error loading <i>assetName</i> . The resource was not a binary resource, or the resource was not found.

See Also

Reference

[ResourceContentManager Class](#)



[ResourceContentManager Members](#)

[Microsoft.Xna.Framework.Content Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceContentManager Properties

Public Properties

	Name	Description
	RootDirectory	(Inherited from ContentManager .)
	ServiceProvider	(Inherited from ContentManager .)

See Also

Reference

[ResourceContentManager Class](#)

[Microsoft.Xna.Framework.Content Namespace](#)









Microsoft.Xna.Framework.Design Namespace

Note

This namespace is available only when developing for Windows.

Provides a unified way of converting types of values to other types.

Classes

Name	Description
 BoundingBoxConverter	Provides a unified way of converting BoundingBox values to other types, as well as for accessing standard values and subproperties.
 BoundingSphereConverter	Provides a unified way of converting BoundingSphere values to other types, as well as for accessing standard values and subproperties.
 ColorConverter	Provides a unified way of converting Color values to other types, as well as for accessing standard values and subproperties.
 MathTypeConverter	Provides a unified way of converting math type values to other types, as well as for accessing standard values and subproperties.
 MatrixConverter	Provides a unified way of converting Matrix values to other types, as well as for accessing standard values and subproperties.
 PlaneConverter	Provides a unified way of converting Plane values to other types, as well as for accessing standard values and subproperties.
 PointConverter	Provides a unified way of converting Point values to other types, as well as for accessing standard values and subproperties.
 QuaternionConverter	Provides a unified way of converting Quaternion values to other types, as well as for accessing standard values and subproperties.
 RayConverter	Provides a unified way of converting Ray values to other types, as well as for accessing standard values and subproperties.
 RectangleConverter	Provides a unified way of converting Rectangle values to other types, as well as for accessing standard values and subproperties.
 Vector2Converter	Provides a unified way of converting Vector2 values to other types, as well as for accessing standard values and subproperties.
 Vector3Converter	Provides a unified way of converting Vector3 values to other types, as well as for accessing standard values and subproperties.
 Vector4Converter	Provides a unified way of converting Vector4 values to other types, as well as for accessing standard values and subproperties.

BoundingBoxConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [BoundingBox](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class BoundingBoxConverter : MathTypeConverter
```

See Also

Reference

[BoundingBoxConverter Members](#)


[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista











BoundingBoxConverter Members

The following tables list the members exposed by the BoundingBoxConverter type.






Public Constructors

Name	Description
 BoundingBoxConverter	Initializes a new instance of the BoundingBoxConverter class.

Public Methods

Name	Description
 CanConvertFrom	(Inherited from TypeConverter .)
 CanConvertTo	(Inherited from TypeConverter .)
 ConvertFrom	Overloaded. Converts the given value to the type of this converter.
 ConvertFromInvariantString	(Inherited from TypeConverter .)
 ConvertFromString	(Inherited from TypeConverter .)
 ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
 ConvertToInvariantString	(Inherited from TypeConverter .)
 ConvertToString	(Inherited from TypeConverter .)
 CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
 Equals	(Inherited from Object .)
 GetCreateInstanceSupported	(Inherited from TypeConverter .)
 GetHashCode	(Inherited from Object .)
 GetProperties	(Inherited from TypeConverter .)
 GetPropertiesSupported	(Inherited from TypeConverter .)
 GetStandardValues	(Inherited from TypeConverter .)
 GetStandardValuesExclusive	(Inherited from TypeConverter .)
 GetStandardValuesSupported	(Inherited from TypeConverter .)
 GetType	(Inherited from Object .)
 IsValid	(Inherited from TypeConverter .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetConvertFromException	(Inherited from TypeConverter .)
 GetConvertToException	(Inherited from TypeConverter .)
 MemberwiseClone	(Inherited from Object .)
 SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter Fields

See Also

Reference

[BoundingBoxConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [BoundingBoxConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingBoxConverter ()
```

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista

BoundingBoxConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
BoundingBoxConverter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
BoundingBoxConverter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
BoundingBoxConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
BoundingBoxConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
BoundingBoxConverter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this BoundingBoxConverter is associated with, using the specified context, given a set of property values for the object.
BoundingBoxConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [BoundingBoxConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [BoundingBox](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class BoundingBoxConverter : MathTypeConverter
```

See Also

Reference

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter Members

The following tables list the members exposed by the BoundingBoxConverter type.

Public Constructors

Name	Description
BoundingBoxConverter	Initializes a new instance of the BoundingBoxConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter Fields

See Also

Reference

[BoundingBoxConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [BoundingBoxConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingBoxConverter ()
```

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
BoundingBoxConverter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
BoundingBoxConverter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
BoundingBoxConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
BoundingBoxConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoundingBoxConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
BoundingBoxConverter.CreateInstance(ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this BoundingBoxConverter is associated with, using the specified context, given a set of property values for the object.
BoundingBoxConverter.CreateInstance(IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

BoundingBoxConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [BoundingBoxConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[BoundingBoxConverter Class](#)

[BoundingBoxConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ColorConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Color](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class ColorConverter : MathTypeConverter
```

See Also

Reference

[ColorConverter Members](#)


[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista











ColorConverter Members

The following tables list the members exposed by the ColorConverter type.






Public Constructors

Name	Description
 ColorConverter	Initializes a new instance of the ColorConverter class.

Public Methods

Name	Description
 CanConvertFrom	(Inherited from TypeConverter .)
 CanConvertTo	(Inherited from TypeConverter .)
 ConvertFrom	Overloaded. Converts the given value to the type of this converter.
 ConvertFromInvariantString	(Inherited from TypeConverter .)
 ConvertFromString	(Inherited from TypeConverter .)
 ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
 ConvertToInvariantString	(Inherited from TypeConverter .)
 ConvertToString	(Inherited from TypeConverter .)
 CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
 Equals	(Inherited from Object .)
 GetCreateInstanceSupported	(Inherited from TypeConverter .)
 GetHashCode	(Inherited from Object .)
 GetProperties	(Inherited from TypeConverter .)
 GetPropertiesSupported	(Inherited from TypeConverter .)
 GetStandardValues	(Inherited from TypeConverter .)
 GetStandardValuesExclusive	(Inherited from TypeConverter .)
 GetStandardValuesSupported	(Inherited from TypeConverter .)
 GetType	(Inherited from Object .)
 IsValid	(Inherited from TypeConverter .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetConvertFromException	(Inherited from TypeConverter .)
 GetConvertToException	(Inherited from TypeConverter .)
 MemberwiseClone	(Inherited from Object .)
 SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[ColorConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

ColorConverter Fields

See Also

Reference

[ColorConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

ColorConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the **ColorConverter** class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ColorConverter ()
```

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ColorConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[ColorConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

ColorConverter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
ColorConverter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given value to the type of this converter.
ColorConverter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

ColorConverter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given value to the type of this converter.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ColorConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
ColorConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
ColorConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

ColorConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SuppressMessageAttribute("Microsoft.Performance", "CA1800")]  
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ColorConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
ColorConverter.CreateInstance (IPropertyDescriptorContext, IDictionary)	Re-creates an object given a set of property values for the object.
ColorConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

ColorConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Re-creates an object given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[ColorConverter Class](#)

[ColorConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting math type values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class MathTypeConverter : ExpandableObjectConverter
```

See Also

Reference

[MathTypeConverter Members](#)


[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista



MathTypeConverter Members

The following tables list the members exposed by the MathTypeConverter type.




















Public Constructors

Name	Description
 MathTypeConverter	Initializes a new instance of the MathTypeConverter class.






Protected Fields

Name	Description
 propertyDescriptions	Represents a collection of PropertyDescriptor objects.
 supportStringConvert	Returns whether string conversion is supported.

Public Methods

Name	Description
 CanConvertFrom	Overloaded. Returns whether this converter can convert an object of one type to the type of this converter.
 CanConvertTo	Overloaded. Returns whether this converter can convert the object to the specified type.
 ConvertFrom	(Inherited from TypeConverter .)
 ConvertFromInvariantString	(Inherited from TypeConverter .)
 ConvertFromString	(Inherited from TypeConverter .)
 ConvertTo	(Inherited from TypeConverter .)
 ConvertToInvariantString	(Inherited from TypeConverter .)
 ConvertToString	(Inherited from TypeConverter .)
 CreateInstance	(Inherited from TypeConverter .)
 Equals	(Inherited from Object .)
 GetCreateInstanceSupported	Overloaded. Returns whether changing a value on this object requires a call to the CreateInstance method to create a new value.
 GetHashCode	(Inherited from Object .)
 GetProperties	Overloaded. Returns a collection of properties for the type of array specified by the value parameter.
 GetPropertiesSupported	Overloaded. Returns whether this object supports properties.
 GetStandardValues	(Inherited from TypeConverter .)
 GetStandardValuesExclusive	(Inherited from TypeConverter .)
 GetStandardValuesSupported	(Inherited from TypeConverter .)
 GetType	(Inherited from Object .)
 IsValid	(Inherited from TypeConverter .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetConvertFromException	(Inherited from TypeConverter .)
 GetConvertToException	(Inherited from TypeConverter .)
 MemberwiseClone	(Inherited from Object .)
 SortProperties	(Inherited from TypeConverter .)

See Also



Reference

[MathTypeConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter Fields

Protected Fields

	Name	Description
	propertyDescriptions	Represents a collection of PropertyDescriptor objects.
	supportStringConvert	Returns whether string conversion is supported.

See Also

Reference

[MathTypeConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.propertyDescriptions Field

Note

This field is available only when developing for Windows.

Represents a collection of [PropertyDescriptor](#) objects.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected PropertyDescriptorCollection propertyDescriptions
```

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter.supportStringConvert Field

Note

This field is available only when developing for Windows.

Returns whether string conversion is supported.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected bool supportStringConvert
```

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [MathTypeConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MathTypeConverter ()
```

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter Methods

Public Methods

Name	Description
CanConvertFrom	Overloaded. Returns whether this converter can convert an object of one type to the type of this converter.
CanConvertTo	Overloaded. Returns whether this converter can convert the object to the specified type.
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	(Inherited from TypeConverter .)
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	(Inherited from TypeConverter .)
Equals	(Inherited from Object .)
GetCreateInstanceSupported	Overloaded. Returns whether changing a value on this object requires a call to the CreateInstance method to create a new value.
GetHashCode	(Inherited from Object .)
GetProperties	Overloaded. Returns a collection of properties for the type of array specified by the value parameter.
GetPropertiesSupported	Overloaded. Returns whether this object supports properties.
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[MathTypeConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.CanConvertFrom Method

Returns whether this converter can convert an object of one type to the type of this converter.

Overload List

Name	Description
MathTypeConverter.CanConvertFrom (ITypeDescriptorContext, Type)	Returns whether this converter can convert an object of the given type to the type of this converter, using the specified context.
MathTypeConverter.CanConvertFrom (Type)	(Inherited from TypeConverter .)

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.CanConvertFrom Method (ITypeDescriptorContext, Type)

Note

This method is available only when developing for Windows.

Returns whether this converter can convert an object of the given type to the type of this converter, using the specified context.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool CanConvertFrom (
    ITypeDescriptorContext context,
    Type sourceType
)
```

Parameters

context

The format context.

sourceType

The type you want to convert from.

Return Value

true if this converter can perform the conversion; **false** otherwise.

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter.CanConvertTo Method

Returns whether this converter can convert the object to the specified type.

Overload List

Name	Description
MathTypeConverter.CanConvertTo (ITypeDescriptorContext, Type)	Returns whether this converter can convert an object of one type to the type of this converter.
MathTypeConverter.CanConvertTo (Type)	(Inherited from TypeConverter .)

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.CanConvertTo Method (ITypeDescriptorContext, Type)

Note

This method is available only when developing for Windows.

Returns whether this converter can convert an object of one type to the type of this converter.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool CanConvertTo (  
    ITypeDescriptorContext context,  
    Type destinationType  
)
```

Parameters

context

The format context.

destinationType

The destination type.

Return Value

true if this converter can perform the conversion; **false** otherwise.

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter.CreateInstanceSupported Method

Returns whether changing a value on this object requires a call to the **CreateInstance** method to create a new value.

Overload List

Name	Description
MathTypeConverter.CreateInstanceSupported (ITypeDescriptorContext)	Returns whether changing a value on this object requires a call to CreateInstance to create a new value, using the specified context.
MathTypeConverter.CreateInstanceSupported ()	(Inherited from TypeConverter .)

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.CreateInstanceSupported Method (ITypeDescriptorContext)

Note

This method is available only when developing for Windows.

Returns whether changing a value on this object requires a call to **CreateInstance** to create a new value, using the specified context.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool GetCreateInstanceSupported (  
    ITypeDescriptorContext context  
)
```

Parameters

context

The format context.

Return Value

true if changing a property on this object requires a call to **CreateInstance** to create a new value; **false** otherwise.

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter.GetProperties Method

Returns a collection of properties for the type of array specified by the value parameter.

Overload List

Name	Description
MathTypeConverter.GetProperties (ITypeDescriptorContext, Object, Attribute[])	Returns a collection of properties for the type of array specified by the value parameter.
MathTypeConverter.GetProperties (ITypeDescriptorContext, Object)	(Inherited from TypeConverter .)
MathTypeConverter.GetProperties (Object)	(Inherited from TypeConverter .)

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.GetProperties Method (ITypeDescriptorContext, Object, Attribute[])

Note

This method is available only when developing for Windows.

Returns a collection of properties for the type of array specified by the value parameter.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override PropertyDescriptorCollection GetProperties (  
    ITypeDescriptorContext context,  
    Object value,  
    Attribute[] attributes  
)
```

Parameters

context

The format context.

value

The type of array for which to get properties.

attributes

An array to use as a filter.

Return Value

The properties that are exposed for this data type, or **null** if there are no properties.

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MathTypeConverter.GetPropertiesSupported Method

Returns whether this object supports properties.

Overload List

Name	Description
MathTypeConverter.GetPropertiesSupported (ITypeDescriptorContext)	Returns whether this object supports properties, using the specified context.
MathTypeConverter.GetPropertiesSupported ()	(Inherited from TypeConverter .)

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MathTypeConverter.GetPropertiesSupported Method (ITypeDescriptorContext)

Note

This method is available only when developing for Windows.

Returns whether this object supports properties, using the specified context.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool GetPropertiesSupported (
    ITypeDescriptorContext context
)
```

Parameters

context

The format context.

Return Value

true if **GetProperties** should be called to find the properties of this object; **false** otherwise.

See Also

Reference

[MathTypeConverter Class](#)

[MathTypeConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MatrixConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Matrix](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class MatrixConverter : MathTypeConverter
```

See Also

Reference

[MatrixConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista

MatrixConverter Members

The following tables list the members exposed by the MatrixConverter type.

Public Constructors

Name	Description
MatrixConverter	Initializes a new instance of the MatrixConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[MatrixConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MatrixConverter Fields

See Also

Reference

[MatrixConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MatrixConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [MatrixConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MatrixConverter ()
```

See Also

Reference

[MatrixConverter Class](#)

[MatrixConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista

MatrixConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[MatrixConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MatrixConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
MatrixConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
MatrixConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[MatrixConverter Class](#)

[MatrixConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MatrixConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[MatrixConverter Class](#)

[MatrixConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MatrixConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
MatrixConverter.CreateInstance (IPropertyDescriptorContext, IDictionary)	Creates an instance of the type that this MatrixConverter is associated with, using the specified context, given a set of property values for the object.
MatrixConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[MatrixConverter Class](#)

[MatrixConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

MatrixConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [MatrixConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[MatrixConverter Class](#)

[MatrixConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PlaneConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Plane](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class PlaneConverter : MathTypeConverter
```

See Also

Reference

[PlaneConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista

PlaneConverter Members

The following tables list the members exposed by the PlaneConverter type.

Public Constructors

Name	Description
PlaneConverter	Initializes a new instance of the PlaneConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[PlaneConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlaneConverter Fields

See Also

Reference

[PlaneConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlaneConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [PlaneConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaneConverter ()
```

See Also

Reference

[PlaneConverter Class](#)

[PlaneConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PlaneConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[PlaneConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlaneConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
PlaneConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
PlaneConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[PlaneConverter Class](#)

[PlaneConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlaneConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[PlaneConverter Class](#)

[PlaneConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PlaneConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
PlaneConverter.CreateInstance (IPropertyDescriptorContext, IDictionary)	Creates an instance of the type that this PlaneConverter is associated with, using the specified context, given a set of property values for the object.
PlaneConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[PlaneConverter Class](#)

[PlaneConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlaneConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [PlaneConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[PlaneConverter Class](#)

[PlaneConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PointConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Point](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class PointConverter : MathTypeConverter
```

See Also

Reference

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PointConverter Members

The following tables list the members exposed by the PointConverter type.

Public Constructors

Name	Description
PointConverter	Initializes a new instance of the PointConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[PointConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PointConverter Fields

See Also

Reference

[PointConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PointConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [PointConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PointConverter ()
```

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PointConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[PointConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PointConverter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
PointConverter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
PointConverter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PointConverter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentException	Invalid string format.

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PointConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
PointConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
PointConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PointConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PointConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
PointConverter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this PointConverter is associated with, using the specified context, given a set of property values for the object.
PointConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PointConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [PointConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (
    ITypeDescriptorContext context,
    IDictionary propertyValues
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[PointConverter Class](#)

[PointConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

QuaternionConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Quaternion](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class QuaternionConverter : MathTypeConverter
```

See Also

Reference

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

QuaternionConverter Members

The following tables list the members exposed by the QuaternionConverter type.

Public Constructors

Name	Description
QuaternionConverter	Initializes a new instance of the QuaternionConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[QuaternionConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

QuaternionConverter Fields

See Also

Reference

[QuaternionConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

QuaternionConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [QuaternionConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public QuaternionConverter ()
```

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista

QuaternionConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[QuaternionConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

QuaternionConverter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
QuaternionConverter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
QuaternionConverter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

QuaternionConverter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentException	Invalid string format.

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

QuaternionConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
QuaternionConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
QuaternionConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

QuaternionConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

QuaternionConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
QuaternionConverter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this QuaternionConverter is associated with, using the specified context, given a set of property values for the object.
QuaternionConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

QuaternionConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [QuaternionConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[QuaternionConverter Class](#)

[QuaternionConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RayConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Ray](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class RayConverter : MathTypeConverter
```

See Also

Reference

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista

RayConverter Members

The following tables list the members exposed by the RayConverter type.

Public Constructors

Name	Description
RayConverter	Initializes a new instance of the RayConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[RayConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RayConverter Fields

See Also

Reference

[RayConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RayConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [RayConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RayConverter ()
```

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RayConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[RayConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RayConverter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
RayConverter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
RayConverter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RayConverter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RayConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
RayConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
RayConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RayConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RayConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
RayConverter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this RayConverter is associated with, using the specified context, given a set of property values for the object.
RayConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RayConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [RayConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[RayConverter Class](#)

[RayConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RectangleConverter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Rectangle](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class RectangleConverter : MathTypeConverter
```

See Also

Reference

[RectangleConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RectangleConverter Members

The following tables list the members exposed by the RectangleConverter type.

Public Constructors

Name	Description
RectangleConverter	Initializes a new instance of the RectangleConverter class.

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[RectangleConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RectangleConverter Fields

See Also

Reference

[RectangleConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RectangleConverter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [RectangleConverter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RectangleConverter ()
```

See Also

Reference

[RectangleConverter Class](#)

[RectangleConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RectangleConverter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	(Inherited from TypeConverter .)
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[RectangleConverter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RectangleConverter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
RectangleConverter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
RectangleConverter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[RectangleConverter Class](#)

[RectangleConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RectangleConverter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[RectangleConverter Class](#)

[RectangleConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RectangleConverter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
RectangleConverter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this RectangleConverter is associated with, using the specified context, given a set of property values for the object.
RectangleConverter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[RectangleConverter Class](#)

[RectangleConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

RectangleConverter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [RectangleConverter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[RectangleConverter Class](#)

[RectangleConverter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector2Converter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Vector2](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class Vector2Converter : MathTypeConverter
```

See Also

Reference

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista
















Vector2Converter Members

The following tables list the members exposed by the Vector2Converter type.






Public Constructors

Name	Description
 Vector2Converter	Initializes a new instance of the Vector2Converter class.

Public Methods

Name	Description
 CanConvertFrom	(Inherited from TypeConverter .)
 CanConvertTo	(Inherited from TypeConverter .)
 ConvertFrom	Overloaded. Converts the given value to the type of this converter.
 ConvertFromInvariantString	(Inherited from TypeConverter .)
 ConvertFromString	(Inherited from TypeConverter .)
 ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
 ConvertToInvariantString	(Inherited from TypeConverter .)
 ConvertToString	(Inherited from TypeConverter .)
 CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
 Equals	(Inherited from Object .)
 GetCreateInstanceSupported	(Inherited from TypeConverter .)
 GetHashCode	(Inherited from Object .)
 GetProperties	(Inherited from TypeConverter .)
 GetPropertiesSupported	(Inherited from TypeConverter .)
 GetStandardValues	(Inherited from TypeConverter .)
 GetStandardValuesExclusive	(Inherited from TypeConverter .)
 GetStandardValuesSupported	(Inherited from TypeConverter .)
 GetType	(Inherited from Object .)
 IsValid	(Inherited from TypeConverter .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetConvertFromException	(Inherited from TypeConverter .)
 GetConvertToException	(Inherited from TypeConverter .)
 MemberwiseClone	(Inherited from Object .)
 SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[Vector2Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector2Converter Fields

See Also

Reference

[Vector2Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector2Converter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [Vector2Converter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2Converter ()
```

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector2Converter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[Vector2Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector2Converter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
Vector2Converter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
Vector2Converter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector2Converter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentException	Invalid string format.

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector2Converter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
Vector2Converter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
Vector2Converter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector2Converter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector2Converter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
Vector2Converter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this Vector2Converter is associated with, using the specified context, given a set of property values for the object.
Vector2Converter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector2Converter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [Vector2Converter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[Vector2Converter Class](#)

[Vector2Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector3Converter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Vector3](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class Vector3Converter : MathTypeConverter
```

See Also

Reference

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista












Vector3Converter Members

The following tables list the members exposed by the Vector3Converter type.






Public Constructors

Name	Description
 Vector3Converter	Initializes a new instance of the Vector3Converter class.

Public Methods

Name	Description
 CanConvertFrom	(Inherited from TypeConverter .)
 CanConvertTo	(Inherited from TypeConverter .)
 ConvertFrom	Overloaded. Converts the given value to the type of this converter.
 ConvertFromInvariantString	(Inherited from TypeConverter .)
 ConvertFromString	(Inherited from TypeConverter .)
 ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
 ConvertToInvariantString	(Inherited from TypeConverter .)
 ConvertToString	(Inherited from TypeConverter .)
 CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
 Equals	(Inherited from Object .)
 GetCreateInstanceSupported	(Inherited from TypeConverter .)
 GetHashCode	(Inherited from Object .)
 GetProperties	(Inherited from TypeConverter .)
 GetPropertiesSupported	(Inherited from TypeConverter .)
 GetStandardValues	(Inherited from TypeConverter .)
 GetStandardValuesExclusive	(Inherited from TypeConverter .)
 GetStandardValuesSupported	(Inherited from TypeConverter .)
 GetType	(Inherited from Object .)
 IsValid	(Inherited from TypeConverter .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetConvertFromException	(Inherited from TypeConverter .)
 GetConvertToException	(Inherited from TypeConverter .)
 MemberwiseClone	(Inherited from Object .)
 SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[Vector3Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector3Converter Fields

See Also

Reference

[Vector3Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector3Converter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [Vector3Converter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3Converter ()
```

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector3Converter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[Vector3Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector3Converter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
Vector3Converter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
Vector3Converter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector3Converter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentException	Invalid string format.

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector3Converter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
Vector3Converter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
Vector3Converter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector3Converter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector3Converter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
Vector3Converter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this Vector3Converter is associated with, using the specified context, given a set of property values for the object.
Vector3Converter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector3Converter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [Vector3Converter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[Vector3Converter Class](#)

[Vector3Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector4Converter Class

Note

This class is available only when developing for Windows.

Provides a unified way of converting [Vector4](#) values to other types, as well as for accessing standard values and subproperties.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class Vector4Converter : MathTypeConverter
```

See Also

Reference

[Vector4Converter Members](#)


[Microsoft.Xna.Framework.Design Namespace](#)

Platforms Windows XP SP2, Windows Vista



















Vector4Converter Members

The following tables list the members exposed by the Vector4Converter type.






Public Constructors

Name	Description
 Vector4Converter	Initializes a new instance of the Vector4Converter class.

Public Methods

Name	Description
 CanConvertFrom	(Inherited from TypeConverter .)
 CanConvertTo	(Inherited from TypeConverter .)
 ConvertFrom	Overloaded. Converts the given value to the type of this converter.
 ConvertFromInvariantString	(Inherited from TypeConverter .)
 ConvertFromString	(Inherited from TypeConverter .)
 ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
 ConvertToInvariantString	(Inherited from TypeConverter .)
 ConvertToString	(Inherited from TypeConverter .)
 CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
 Equals	(Inherited from Object .)
 GetCreateInstanceSupported	(Inherited from TypeConverter .)
 GetHashCode	(Inherited from Object .)
 GetProperties	(Inherited from TypeConverter .)
 GetPropertiesSupported	(Inherited from TypeConverter .)
 GetStandardValues	(Inherited from TypeConverter .)
 GetStandardValuesExclusive	(Inherited from TypeConverter .)
 GetStandardValuesSupported	(Inherited from TypeConverter .)
 GetType	(Inherited from Object .)
 IsValid	(Inherited from TypeConverter .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetConvertFromException	(Inherited from TypeConverter .)
 GetConvertToException	(Inherited from TypeConverter .)
 MemberwiseClone	(Inherited from Object .)
 SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[Vector4Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector4Converter Fields

See Also

Reference

[Vector4Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector4Converter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [Vector4Converter](#) class.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4Converter ()
```

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector4Converter Methods

Public Methods

Name	Description
CanConvertFrom	(Inherited from TypeConverter .)
CanConvertTo	(Inherited from TypeConverter .)
ConvertFrom	Overloaded. Converts the given value to the type of this converter.
ConvertFromInvariantString	(Inherited from TypeConverter .)
ConvertFromString	(Inherited from TypeConverter .)
ConvertTo	Overloaded. Converts the given value object to the specified type, using the specified context and culture information.
ConvertToInvariantString	(Inherited from TypeConverter .)
ConvertToString	(Inherited from TypeConverter .)
CreateInstance	Overloaded. Re-creates an object given a set of property values for the object.
Equals	(Inherited from Object .)
GetCreateInstanceSupported	(Inherited from TypeConverter .)
GetHashCode	(Inherited from Object .)
GetProperties	(Inherited from TypeConverter .)
GetPropertiesSupported	(Inherited from TypeConverter .)
GetStandardValues	(Inherited from TypeConverter .)
GetStandardValuesExclusive	(Inherited from TypeConverter .)
GetStandardValuesSupported	(Inherited from TypeConverter .)
GetType	(Inherited from Object .)
IsValid	(Inherited from TypeConverter .)
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetConvertFromException	(Inherited from TypeConverter .)
GetConvertToException	(Inherited from TypeConverter .)
MemberwiseClone	(Inherited from Object .)
SortProperties	(Inherited from TypeConverter .)

See Also

Reference

[Vector4Converter Class](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector4Converter.ConvertFrom Method

Converts the given value to the type of this converter.

Overload List

Name	Description
Vector4Converter.ConvertFrom (ITypeDescriptorContext, CultureInfo, Object)	Converts the given object to the type of this converter, using the specified context and culture information.
Vector4Converter.ConvertFrom (Object)	(Inherited from TypeConverter .)

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector4Converter.ConvertFrom Method (ITypeDescriptorContext, CultureInfo, Object)

Note

This method is available only when developing for Windows.

Converts the given object to the type of this converter, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertFrom (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value  
)
```

Parameters

context

The format context.

culture

The current culture.

value

The object to convert.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentException	Invalid string format.

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector4Converter.ConvertTo Method

Converts the given value object to the specified type, using the specified context and culture information.

Overload List

Name	Description
Vector4Converter.ConvertTo (ITypeDescriptorContext, CultureInfo, Object, Type)	Converts the given value object to the specified type, using the specified context and culture information.
Vector4Converter.ConvertTo (Object, Type)	(Inherited from TypeConverter .)

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector4Converter.ConvertTo Method (ITypeDescriptorContext, CultureInfo, Object, Type)

Note

This method is available only when developing for Windows.

Converts the given value object to the specified type, using the specified context and culture information.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object ConvertTo (  
    ITypeDescriptorContext context,  
    CultureInfo culture,  
    Object value,  
    Type destinationType  
)
```

Parameters

context

The format context.

culture

The culture to use in the conversion.

value

The object to convert.

destinationType

The destination type.

Return Value

The converted value.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationType</i> is null .

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Vector4Converter.CreateInstance Method

Re-creates an object given a set of property values for the object.

Overload List

Name	Description
Vector4Converter.CreateInstance (ITypeDescriptorContext, IDictionary)	Creates an instance of the type that this Vector4Converter is associated with, using the specified context, given a set of property values for the object.
Vector4Converter.CreateInstance (IDictionary)	(Inherited from TypeConverter .)

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)

[Microsoft.Xna.Framework.Design Namespace](#)

Vector4Converter.CreateInstance Method (ITypeDescriptorContext, IDictionary)

Note

This method is available only when developing for Windows.

Creates an instance of the type that this [Vector4Converter](#) is associated with, using the specified context, given a set of property values for the object.

Namespace: Microsoft.Xna.Framework.Design

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Object CreateInstance (  
    ITypeDescriptorContext context,  
    IDictionary propertyValues  
)
```

Parameters

context

The format context.

propertyValues

The new property values.

Return Value

An object representing *propertyValues*, or **null** if the object cannot be created.

See Also

Reference

[Vector4Converter Class](#)

[Vector4Converter Members](#)






















[Microsoft.Xna.Framework.Design Namespace](#)

PlatformsWindows XP SP2, Windows Vista



Microsoft.Xna.Framework.GamerServices Namespace

Contains classes that implement various services related to gamers. These services communicate directly with the gamer, the gamer's data, or otherwise reflect choices the gamer makes. Gamer services include input device and profile data APIs.





Classes











Name	Description
 AvatarAnimation	Provides methods and properties for animating an avatar using standard animations (for example, celebrate).
 AvatarDescription	Provides access to the methods and properties of the description data for an avatar.
 AvatarRenderer	Provides properties and methods for rendering a standard avatar.
 FriendCollection	Represents the complete friends list of a local gamer.
 FriendGamer	Provides the presence information of a friend of the local gamer.
 GameDefaults	Describes a gamer's preferred settings.
 Gamer	Abstract base class for types that represent game players (profiles that have an associated gamertag). The concrete types SignedInGamer and NetworkGamer derive from this.
 GamerCollection	Represents a collection of gamers. This collection cannot be modified and is updated automatically during the call to Update .
 GamerPresence	Provides properties to set the rich presence state for a locally signed-in gamer profile.
 GamerPrivilegeException	Thrown if a gamer services or multiplayer API is called without a valid, signed-in profile.
 GamerPrivileges	Describes what operations a gamer is allowed to perform.
 GamerProfile	Profile settings describing information about a gamer such as the gamer's motto, reputation, and gamer picture. This data is accessible for both locally signed in profiles and remote gamers that you are playing with in a multiplayer session.
 GamerServicesComponent	Wraps the functionality of the GamerServicesDispatcher .
 GamerServicesDispatcher	Implements the Windows-specific portion of a GamerServicesDispatcher class.
 GamerServicesNotAvailableException	Thrown if the gamer services system cannot be successfully initialized.
 Guide	Provides access to the Guide user interface.
 GuideAlreadyVisibleException	Thrown if an attempt is made to display a component of the Guide user interface when a Guide component is already displayed.
 SignedInEventArgs	Represents the arguments passed to a SignedIn event.
 SignedInGamer	Represents a gamer (a profile that has an associated gamertag) on the local system.
 SignedInGamerCollection	Represents a collection of gamers on the local system.
 SignedOutEventArgs	Represents the arguments passed to a SignedOut event.

Structures

Name	Description
 AvatarExpression	Contains the various components of the avatar's face, such as the left and right eyebrows.
 GamerCollection.GamerCollectionEnumerator	Provides the ability to iterate through the gamers in an GamerCollection .

Enumerations

Name	Description
 AvatarAnimationPreset	Defines standard animations for avatars.
 AvatarBodyType	Defines the body type of the avatar.
 AvatarBone	Defines a list of the useful bones of the avatar model.
 AvatarEye	Defines the standard animation textures for an avatar's eyes.

 AvatarEyebrow	Defines the standard animation textures for an avatar's eyebrows.
 AvatarMouth	Defines the standard animation textures for an avatar's mouth.
 ControllerSensitivity	Indicates how sensitive this gamer prefers controller input to be.
 GameDifficulty	Indicates how difficult this gamer likes things to be.
 GamerPresenceMode	Settings defining the status string that will appear when you view a friend through the Xbox LIVE Guide or on Xbox.com. Use the PresenceMode property to set this option.
 GamerPrivilegeSetting	Describes the conditions in which a privilege is available.
 GamerZone	This style of social gaming preferred by this Xbox LIVE member.
 MessageBoxIcon	Defines the different icons for a message box.
 NotificationPosition	Determines where notifications appear on the screen.
 RacingCameraAngle	Indicates which camera angle this gamer prefers to use in racing games.

Remarks

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

See Also

Concepts

[Gamer Services Overview](#)

AvatarAnimation Class

Provides methods and properties for animating an avatar using standard animations (for example, celebrate). For more information on the standard animations available to avatars, see [AvatarAnimationPreset](#).

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class AvatarAnimation : IDisposable
```

Remarks

Use this class, in conjunction with [AvatarRenderer](#), to provide standard animations for the avatar model. For more information, see [How To: Render and Animate an Avatar Using AvatarRenderer..](#)

See Also

Reference

[AvatarAnimation Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






AvatarAnimation Members

The following tables list the members exposed by the AvatarAnimation type.








Public Constructors

Name	Description
 AvatarAnimation	Creates a new instance of AvatarAnimation, and initializes it with the specified animation.



Public Properties

Name	Description
 BoneTransforms	Gets the current position of the bones at the time specified by CurrentPosition .
 CurrentPosition	Gets or sets the current time position in the animation.
 Expression	Gets the expression of the related animation at the current time position.
 IsDisposed	Gets the disposed state of the avatar animation.
 Length	Gets the length of the current animation.

Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Update	Updates the current time position of the avatar animation.

Protected Methods

Name	Description
 Finalize	Allows the avatar animation to attempt to free resources, and it performs other cleanup operations before garbage collection reclaims the animation.
 Memberwise Clone	(Inherited from Object .)

See Also

Reference

[AvatarAnimation Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarAnimation Constructor

Creates a new instance of **AvatarAnimation**, and initializes it with the specified animation.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarAnimation (  
    AvatarAnimationPreset animationPreset  
)
```

Parameters

animationPreset

The specified standard animation.

See Also

Reference

[AvatarAnimation Class](#)








[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Update	Updates the current time position of the avatar animation.

Protected Methods

	Name	Description
	Finalize	Allows the avatar animation to attempt to free resources, and it performs other cleanup operations before garbage collection reclaims the animation.
	Memberwise Clone	(Inherited from Object .)

See Also

Reference

[AvatarAnimation Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarAnimation.Dispose Method

Immediately releases the unmanaged resources used by this object.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Overload List

Name	Description
AvatarAnimation.Dispose ()	Immediately releases the unmanaged resources used by this object.
AvatarAnimation.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarAnimation.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.Finalize Method

Allows the avatar animation to attempt to free resources, and it performs other cleanup operations before garbage collection reclaims the animation.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.Update Method

Updates the current time position of the avatar animation.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Update (  
    TimeSpan elapsedAnimationTime,  
    bool loop  
)
```

Parameters

elapsedAnimationTime

Elapsed time since the last animation frame. This value can be either a positive or negative time span. If the value is negative, animation playback is reversed.

loop

true if the animation playback is to be looped; otherwise, **false**.

See Also

Reference

[AvatarAnimation Class](#)






[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation Properties

Public Properties

	Name	Description
	BoneTransforms	Gets the current position of the bones at the time specified by CurrentPosition .
	CurrentPosition	Gets or sets the current time position in the animation.
	Expression	Gets the expression of the related animation at the current time position.
	IsDisposed	Gets the disposed state of the avatar animation.
	Length	Gets the length of the current animation.

See Also

Reference

[AvatarAnimation Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarAnimation.BoneTransforms Property

Gets the current position of the bones at the time specified by [CurrentPosition](#). The positions are in local bone space relative to their parent bones.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<Matrix> BoneTransforms { get; }
```

Property Value

Collection of bone transforms.

Remarks

You can pass this matrix list to [AvatarRenderer.Draw](#) using the *bones* parameter. This allows the developer to modify the position of one or more avatar bones before rendering the model.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.CurrentPosition Property

Gets or sets the current time position in the animation.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan CurrentPosition { get; set; }
```

Property Value

Current time position of the animation.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.Expression Property

Gets the expression of the related animation at the current time position.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarExpression Expression { get; }
```

Property Value

Expression of the avatar.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.IsDisposed Property

Gets the disposed state of the avatar animation.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the animation has been disposed; otherwise, **false**.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimation.Length Property

Gets the length of the current animation.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan Length { get; }
```

Property Value

Length of the animation.

See Also

Reference

[AvatarAnimation Class](#)

[AvatarAnimation Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarAnimationPreset Enumeration

Defines standard animations for avatars.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AvatarAnimationPreset
```

Members

Member name	Description
Celebrate	Gender-neutral, celebrating.
Clap	Gender-neutral, applauding.
FemaleAngry	Female, angry.
FemaleConfused	Female, confused.
FemaleCry	Female, cry.
FemaleIdleCheckNails	Female, checking nails.
FemaleIdleFixShoe	Female, fixing shoe.
FemaleIdleLookAround	Female, looking around.
FemaleIdleShiftWeight	Female, shifting weight from one foot to another.
FemaleLaugh	Female, laughing.
FemaleShocked	Female, shocked or surprised.
FemaleYawn	Female, yawning.
MaleAngry	Male, angry.
MaleConfused	Male, confused.
MaleCry	Male, crying.
MaleIdleCheckHand	Male, checking hand.
MaleIdleLookAround	Male, looking around.
MaleIdleShiftWeight	Male, shifting weight from one foot to another.
MaleIdleStretch	Male, stretching.
MaleLaugh	Male, laughing.
MaleSurprised	Male, surprised.
MaleYawn	Male, yawning.
Stand0	Gender-neutral, standing, variation 0.
Stand1	Gender-neutral, standing, variation 1.
Stand2	Gender-neutral, standing, variation 2.
Stand3	Gender-neutral, standing, variation 3.
Stand4	Gender-neutral, standing, variation 4.
Stand5	Gender-neutral, standing, variation 5.
Stand6	Gender-neutral, standing, variation 6.
Stand7	Gender-neutral, standing, variation 7.
Wave	Gender-neutral, waving.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarBodyType Enumeration

Defines the body type of the avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AvatarBodyType
```

Members

Member name	Description
Female	Female avatar.
Male	Male avatar.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarBone Enumeration

Defines a list of the useful bones of the avatar model. This is not a complete list of avatar bones. Some values are not listed. Unlisted bone indices are special and used only when building an animation in an animation package.

Bone indices are sorted by depth value.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AvatarBone
```

Members

Member name	Value	Description
AnkleLeft	11	Left ankle.
AnkleRight	15	Right ankle.
BackLower	1	Lower back.
BackUpper	5	Upper back.
CollarLeft	12	Left collar.
CollarRight	16	Right collar
ElbowLeft	25	Left elbow
ElbowRight	28	Right elbow
FingerIndex2Left	51	Left index finger, 2nd joint.
FingerIndex2Right	56	Right index finger, 2nd joint.
FingerIndex3Left	61	Left index finger, 3rd joint.
FingerIndex3Right	66	Right index finger, 3rd joint.
FingerIndexLeft	37	Left index finger, 1st joint.
FingerIndexRight	44	Right index finger, 1st joint.
FingerMiddle2Left	52	Left middle finger, 2nd joint.
FingerMiddle2Right	57	Right middle finger, 2nd joint.
FingerMiddle3Left	62	Left middle finger, 3rd joint.
FingerMiddle3Right	67	Right middle finger, 3rd joint.
FingerMiddleLeft	38	Left middle finger, 1st joint.
FingerMiddleRight	45	Right middle finger, 1st joint.
FingerRing2Left	53	Left ring finger, 2nd joint.
FingerRing2Right	58	Right ring finger, 2nd joint.
FingerRing3Left	63	Left ring finger, 3rd joint.
FingerRing3Right	68	Right ring finger, 3rd joint.
FingerRingLeft	39	Left ring finger, 1st joint.
FingerRingRight	46	Right ring finger, 1st joint.
FingerSmall2Left	54	Left pinky finger, 2nd joint.
FingerSmall2Right	59	Right pinky finger, 2nd joint.
FingerSmall3Left	64	Left pinky finger, 3rd joint.
FingerSmall3Right	69	Right pinky finger, 3rd joint.
FingerSmallLeft	40	Left pinky finger, 1st joint.
FingerSmallRight	47	Right pinky finger, 2nd joint.
FingerThumb2Left	55	Left thumb, 2nd joint.
FingerThumb2Right	60	Right thumb, 2nd joint.

FingerThumb3Left	65	Left thumb, 3rd joint.
FingerThumb3Right	70	Right thumb, 3rd joint.
FingerThumbLeft	43	Left thumb, 1st joint.
FingerThumbRight	50	Right thumb, 1st joint.
Head	19	Head
HipLeft	2	Left hip.
HipRight	3	Right hip.
KneeLeft	6	Left knee.
Neck	14	Neck.
PropLeft	41	A separate object held in the left hand.
PropRight	48	A separate object held in the left hand.
KneeRight	8	Right knee.
Root	0	Root bone of the avatar skeleton.
ShoulderLeft	20	Left shoulder.
ShoulderRight	22	Right shoulder.
SpecialLeft	42	A special bone located near the left hand of the avatar model.
SpecialRight	49	A special bone located near the right hand of the avatar model.
ToeLeft	21	Left toe.
ToeRight	23	Right toe.
WristLeft	33	Left wrist.
WristRight	36	Right wrist.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription Class

Provides access to the methods and properties of the description data for an avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class AvatarDescription
```

Remarks

The **AvatarDescription** object contains an internal data buffer used by the avatar renderer when loading and animating a gamer's avatar. There are also methods for getting the height and gender of the avatar. Use the [Avatar](#) property of a signed-in gamer to retrieve the related **AvatarDescription** object.

AvatarDescription also provides a number of methods that you can use to create a new random **AvatarDescription** object or to retrieve one from an existing avatar description buffer.

See Also

Reference

[AvatarDescription Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





AvatarDescription Members

The following tables list the members exposed by the AvatarDescription type.








Public Constructors

	Name	Description
	AvatarDescription	Creates a new instance of AvatarDescription using an existing data buffer.



Public Properties

	Name	Description
	BodyType	Gets the body type of the avatar based on the description data.
	Description	Internal description buffer of the avatar, stored as a byte buffer.
	Height	Height of the avatar, from the feet to the top of the head.
	IsValid	Determines whether the internal data buffer is valid.

Public Methods

	Name	Description
	 CreateRandom	Overloaded. Creates an AvatarDescription object with random gender, features, and clothing.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AvatarDescription Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarDescription Constructor

Creates a new instance of **AvatarDescription** using an existing data buffer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarDescription (  
    byte[] data  
)
```

Parameters

data

Existing buffer used to store the information of the newly-created avatar.

Use the buffer specified by the [Description](#) property of a valid [AvatarDescription](#) object. Common scenarios include the recreation of random avatars that were created earlier or when avatar description data is sent over a network to all signed-in gamers. Once that data is received, local avatar descriptions and renderers are created for each remote gamer avatar.

Results are undefined if buffers from other sources are used.

See Also

Reference

[AvatarDescription Class](#)








[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription Methods

Public Methods

	Name	Description
	 CreateRandom	Overloaded. Creates an AvatarDescription object with random gender, features, and clothing.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AvatarDescription Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarDescription.CreateRandom Method

Creates an [AvatarDescription](#) object with random gender, features, and clothing.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Overload List

Name	Description
AvatarDescription.CreateRandom ()	Creates an avatar of random gender, features, and clothing.
AvatarDescription.CreateRandom (AvatarBodyType)	Creates an avatar of the specified body type with random features and clothing.

See Also

Reference

[AvatarDescription Class](#)

[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarDescription.CreateRandom Method ()

Creates an avatar of random gender, features, and clothing.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AvatarDescription CreateRandom ()
```

Return Value

Description of the created avatar.

See Also

Reference

[AvatarDescription Class](#)

[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription.CreateRandom Method (AvatarBodyType)

Creates an avatar of the specified body type with random features and clothing.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AvatarDescription CreateRandom (  
    AvatarBodyType bodyType  
)
```

Parameters

bodyType

Body type of the randomly-created avatar.

Return Value

Description of the created avatar.

Remarks

It is possible that a newly-created random [AvatarDescription](#) object could be an invalid avatar description buffer. For instance, the related [AvatarDescription](#) object was created by a call to **CreateRandom**, from within a Windows application.

See Also

Reference

[AvatarDescription Class](#)





[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription Properties

Public Properties

	Name	Description
	BodyType	Gets the body type of the avatar based on the description data.
	Description	Internal description buffer of the avatar, stored as a byte buffer.
	Height	Height of the avatar, from the feet to the top of the head.
	IsValid	Determines whether the internal data buffer is valid.

See Also

Reference

[AvatarDescription Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarDescription.BodyType Property

Gets the body type of the avatar based on the description data.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarBodyType BodyType { get; }
```

Property Value

Body type of the avatar.

See Also

Reference

[AvatarDescription Class](#)

[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription.Description Property

Internal description buffer of the avatar, stored as a byte buffer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte[] Description { get; }
```

Property Value

Description buffer of the avatar.

See Also

Reference

[AvatarDescription Class](#)

[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription.Height Property

Height of the avatar, from the feet to the top of the head.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Height { get; }
```

Property Value

Height, in meters, of the avatar.

See Also

Reference

[AvatarDescription Class](#)

[AvatarDescription Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarDescription.IsValid Property

Determines whether the internal data buffer is valid.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsValid { get; }
```

Property Value

true if the related [AvatarDescription](#) object contains a valid avatar description, usable by an [AvatarRenderer](#) object; otherwise, **false**.

IsValid always returns **false** on Windows because currently there is no Windows-based avatar implementation.

Remarks

This property is provided only as a quick verification method for developers that receive an [AvatarDescription](#) object from the XNA framework. For example, a value of **false** could indicate that the user does not have an avatar.

See Also

Reference

[AvatarDescription Class](#)

[AvatarDescription Members](#)

[AvatarDescription](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarExpression Structure

Contains the various components of the avatar's face, such as the left and right eyebrows.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct AvatarExpression
```

See Also

Reference

[AvatarExpression Members](#)






[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista




AvatarExpression Members

The following tables list the members exposed by the AvatarExpression type.



Public Properties

Name	Description
 LeftEye	Gets or sets the current texture for the avatar's left eye.
 LeftEyebrow	Gets or sets the current texture for the avatar's left eye.
 Mouth	Gets or sets the current texture for the avatar's mouth.
 RightEye	Gets or sets the current texture for the avatar's right eye.
 RightEyebrow	Gets or sets the current texture for the avatar's right eyebrow.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also




Reference

[AvatarExpression Structure](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarExpression Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[AvatarExpression Structure](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarExpression Properties

Public Properties

	Name	Description
	LeftEye	Gets or sets the current texture for the avatar's left eye.
	LeftEyebrow	Gets or sets the current texture for the avatar's left eye.
	Mouth	Gets or sets the current texture for the avatar's mouth.
	RightEye	Gets or sets the current texture for the avatar's right eye.
	RightEyebrow	Gets or sets the current texture for the avatar's right eyebrow.

See Also

Reference

[AvatarExpression Structure](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarExpression.LeftEye Property

Gets or sets the current texture for the avatar's left eye.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarEye LeftEye { get; set; }
```

Property Value

Texture of the avatar's left eye.

See Also

Reference

[AvatarExpression Structure](#)

[AvatarExpression Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarExpression.LeftEyebrow Property

Gets or sets the current texture for the avatar's left eye.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarEyebrow LeftEyebrow { get; set; }
```

Property Value

Texture of the avatar's left eyebrow.

See Also

Reference

[AvatarExpression Structure](#)

[AvatarExpression Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarExpression.Mouth Property

Gets or sets the current texture for the avatar's mouth.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarMouth Mouth { get; set; }
```

Property Value

Texture of the avatar's mouth.

See Also

Reference

[AvatarExpression Structure](#)

[AvatarExpression Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarExpression.RightEye Property

Gets or sets the current texture for the avatar's right eye.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarEye RightEye { get; set; }
```

Property Value

Texture of the avatar's right eye.

See Also

Reference

[AvatarExpression Structure](#)

[AvatarExpression Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarExpression.RightEyebrow Property

Gets or sets the current texture for the avatar's right eyebrow.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarEyebrow RightEyebrow { get; set; }
```

Property Value

Texture of the avatar's right eyebrow.

See Also

Reference

[AvatarExpression Structure](#)

[AvatarExpression Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarEye Enumeration

Defines the standard animation textures for an avatar's eyes.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AvatarEye
```

Members

Member name	Description
Angry	Angry eye position.
Blink	Blinking eye position.
Confused	Confused eye position.
Happy	Happy eye position.
Laughing	Laughing eye position.
LookDown	Looking down eye position.
LookLeft	Looking left position.
LookRight	Looking right eye position.
LookUp	Looking up eye position.
Neutral	Neutral eye position.
Sad	Sad eye position.
Shocked	Shocked eye position.
Sleeping	Sleeping eye position.
Yawning	Yawning eye position.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarEyebrow Enumeration

Defines the standard animation textures for an avatar's eyebrows.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AvatarEyebrow
```

Members

Member name	Description
Angry	Angry eyebrow position.
Confused	Confused eyebrow position.
Neutral	Neutral eyebrow position.
Raised	Raised eyebrow position.
Sad	Sad eyebrow position.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarMouth Enumeration

Defines the standard animation textures for an avatar's mouth.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum AvatarMouth
```

Members

Member name	Description
Angry	Angry mouth position.
Confused	Confused mouth position.
Happy	Happy mouth position.
Laughing	Laughing mouth position.
Neutral	Neutral mouth position.
PhoneticAi	Phonetic "ai" mouth position (for lip sync).
PhoneticDth	Phonetic "dth" mouth position (for lip sync).
PhoneticEe	Phonetic "ee" mouth position (for lip sync).
PhoneticFv	Phonetic "fv" mouth position (for lip sync).
PhoneticL	Phonetic "l" mouth position (for lip sync).
PhoneticO	Phonetic "o" mouth position (for lip sync).
PhoneticW	Phonetic "w" mouth position (for lip sync).
Sad	Sad mouth position.
Shocked	Shocked mouth position.

See Also

Reference

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer Class

Provides properties and methods for rendering a standard avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class AvatarRenderer : IDisposable
```

Remarks

Use this class, in conjunction with [AvatarAnimation](#), to render an avatar model using standard animations. For more information, see [How To: Render and Animate an Avatar Using AvatarRenderer..](#)

See Also

Reference

[AvatarRenderer Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


AvatarRenderer Members

The following tables list the members exposed by the AvatarRenderer type.











Public Constructors

Name	Description
 AvatarRenderer	Overloaded. Creates a new instance of AvatarRenderer.








Public Fields

Name	Description
 BoneCount	Number of bones in the avatar model.



Public Properties

Name	Description
 AmbientLightColor	Gets or sets the color of the ambient light used by the avatar renderer.
 BindPose	Gets the collection of bind pose positions for each bone of the avatar model.
 IsDisposed	Gets the disposed state of the avatar renderer.
 IsLoaded	Gets the loaded state of the avatar.
 LightColor	Gets or sets the color of the directional light used by the avatar renderer.
 LightDirection	Gets or sets the color of the directional light used by the avatar renderer.
 ParentBones	Collection of the parent indices for each bone in the related BindPose collection.
 Projection	Gets or sets the projection matrix for the avatar.
 View	Gets or sets the view matrix for the avatar.
 World	Gets or sets the world matrix for the avatar.

Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Draw	Draws the avatar to the current render target.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows the avatar renderer to attempt to free resources and perform other cleanup operations before garbage collection reclaims the renderer.
 Memberwise Clone	(Inherited from Object .)

See Also


Reference

[AvatarRenderer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarRenderer Fields

Public Fields

	Name	Description
	BoneCount	Number of bones in the avatar model.

See Also

Reference

[AvatarRenderer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarRenderer.BoneCount Field

Number of bones in the avatar model.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int BoneCount
```

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer Constructor

Creates a new instance of **AvatarRenderer**.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Overload List

Name	Description
AvatarRenderer (AvatarDescription)	Creates a new instance of AvatarRenderer with the specified description.
AvatarRenderer (AvatarDescription, Boolean)	Creates a new instance of AvatarRenderer with the specified description.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarRenderer Constructor (AvatarDescription)

Creates a new instance of **AvatarRenderer** with the specified description.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarRenderer (  
    AvatarDescription avatarDescription  
)
```

Parameters

avatarDescription

Description of the avatar to be rendered.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer Constructor (AvatarDescription, Boolean)

Creates a new instance of **AvatarRenderer** with the specified description. The standard loading effect can also be requested. This effect is animated, indicating that the avatar currently is loading.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.



Figure 1. Avatar loading effect

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarRenderer (  
    AvatarDescription avatarDescription,  
    bool useLoadingEffect  
)
```

Parameters

avatarDescription

Description of the avatar to be rendered.

useLoadingEffect

true if the standard effect is to be used during loading; otherwise, **false**.

See Also

Reference

[AvatarRenderer Class](#)








[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Draw	Draws the avatar to the current render target.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows the avatar renderer to attempt to free resources and perform other cleanup operations before garbage collection reclaims the renderer.
	Memberwise Clone	(Inherited from Object .)

See Also

Reference

[AvatarRenderer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarRenderer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Overload List

Name	Description
AvatarRenderer.Dispose ()	Immediately releases the unmanaged resources used by this object.
AvatarRenderer.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarRenderer.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.Draw Method

Draws the avatar to the current render target.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (  
    IList<Matrix> bones,  
    AvatarExpression expression  
)
```

Parameters

bones

Collection of current bone transforms for the avatar. The transforms are in local bone space relative to their parent.

expression

Current expression textures for the avatar.

Remarks

If any avatar assets are not loaded, the standard loading effect is displayed (determined by the renderer's [constructor](#) call). Determine the current loading state by checking the value of the [IsLoaded](#) property.

The avatar model is always rendered as if [FillMode.Solid](#) is the current fill mode value.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.Finalize Method

Allows the avatar renderer to attempt to free resources and perform other cleanup operations before garbage collection reclaims the renderer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[AvatarRenderer Class](#)











[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer Properties

Public Properties

	Name	Description
	AmbientLightColor	Gets or sets the color of the ambient light used by the avatar renderer.
	BindPose	Gets the collection of bind pose positions for each bone of the avatar model.
	IsDisposed	Gets the disposed state of the avatar renderer.
	IsLoaded	Gets the loaded state of the avatar.
	LightColor	Gets or sets the color of the directional light used by the avatar renderer.
	LightDirection	Gets or sets the color of the directional light used by the avatar renderer.
	ParentBones	Collection of the parent indices for each bone in the related BindPose collection.
	Projection	Gets or sets the projection matrix for the avatar.
	View	Gets or sets the view matrix for the avatar.
	World	Gets or sets the world matrix for the avatar.

See Also

Reference

[AvatarRenderer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

AvatarRenderer.AmbientLightColor Property

Gets or sets the color of the ambient light used by the avatar renderer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 AmbientLightColor { get; set; }
```

Property Value

Color of the ambient light.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.BindPose Property

Gets the collection of bind pose positions for each bone of the avatar model. These positions are in local space, and are relative to the parent bone.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<Matrix> BindPose { get; }
```

Property Value

Collection of bind pose positions for the bones of the avatar model.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.IsDisposed Property

Gets the disposed state of the avatar renderer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the renderer has been disposed; otherwise **false**.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.IsLoaded Property

Gets the loaded state of the avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsLoaded { get; }
```

Property Value

true if the avatar is loaded; otherwise, **false**.

Remarks

Loading an avatar requires some time for accessing the console's hard drive. It is not instantaneous. However, as a rule, it is not a long period of time.

During this time, the standard loading effect is displayed. For more information, see [AvatarRenderer Constructor \(AvatarDescription, Boolean\)](#).

Note

Calling **IsLoaded** or the [Draw](#) method updates the loading status.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.LightColor Property

Gets or sets the color of the directional light used by the avatar renderer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 LightColor { get; set; }
```

Property Value

Color of the directional light.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.LightDirection Property

Gets or sets the color of the directional light used by the avatar renderer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 LightDirection { get; set; }
```

Property Value

Vector of the directional light.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.ParentBones Property

Collection of the parent indices for each bone in the related [BindPose](#) collection.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<int> ParentBones { get; }
```

Property Value

Collection of parent bone indices.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.Projection Property

Gets or sets the projection matrix for the avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix Projection { get; set; }
```

Property Value

Projection matrix used by the avatar renderer.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.View Property

Gets or sets the view matrix for the avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix View { get; set; }
```

Property Value

View matrix used by the avatar renderer.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

AvatarRenderer.World Property

Gets or sets the world matrix for the avatar.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix World { get; set; }
```

Property Value

World matrix used by the avatar renderer.

See Also

Reference

[AvatarRenderer Class](#)

[AvatarRenderer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ControllerSensitivity Enumeration

Indicates how sensitive this gamer prefers controller input to be.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ControllerSensitivity
```

Members

Member name	Description
High	Highly sensitive controller input is preferred.
Low	Below average sensitivity is preferred.
Medium	Average controller sensitivity is preferred.

See Also

Reference

[GameDefaults.ControllerSensitivity Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendCollection Class

Represents the complete friends list of a local gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

```
C#  
public sealed class FriendCollection : GamerCollection<FriendGamer>, IDisposable
```

See Also

Reference

[FriendCollection Members](#)




[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendCollection Members

The following tables list the members exposed by the FriendCollection type.











Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	Allows this object to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object.)

See Also











Reference

[FriendCollection Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

FriendCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection .)
	CopyTo	(Inherited from ReadOnlyCollection .)
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from ReadOnlyCollection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from ReadOnlyCollection .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FriendCollection Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

FriendCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks Call **Dispose** when you are finished using the [FriendCollection](#). The **Dispose** method leaves the [FriendCollection](#) in an unusable state. After calling **Dispose**, you must release all references to the [FriendCollection](#) so the garbage collector can reclaim the memory that the [FriendCollection](#) was occupying.

Note

Always call **Dispose** before you release your last reference to the [FriendCollection](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [FriendCollection](#) object's [Finalize](#) method.

See Also

Reference

[FriendCollection Class](#)

[FriendCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendCollection.Finalize Method

Allows this object to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks This method overrides [Finalize](#). Application code should not call this method; an object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[FriendCollection Class](#)




[FriendCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[FriendCollection Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

FriendCollection.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[FriendCollection Class](#)

[FriendCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer Class

Provides the presence information of a friend of the local gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class FriendGamer : Gamer
```

See Also

Reference

[FriendGamer Members](#)














[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








FriendGamer Members

The following tables list the members exposed by the FriendGamer type.



Public Properties

Name	Description
 FriendRequestReceivedFrom	Gets whether the local gamer who requested the friends list has received a friend request from this gamer.
 FriendRequestSentTo	Gets whether the local gamer who requested the friends list has sent a friend request to this gamer.
 Gamertag	(Inherited from Gamer .)
 HasVoice	Gets whether this friend currently has voice capability.
 InviteAccepted	Gets whether this friend has accepted an invitation from the local gamer who requested the friends list.
 InviteReceivedFrom	Gets whether the local gamer who requested the friends list has received an invitation from this friend.
 InviteRejected	Gets whether this friend has rejected an invitation from the local gamer who requested the friends list.
 InviteSentTo	Gets whether the local gamer who requested the friends list has sent an invitation to this friend.
 IsAway	Gets whether this friend is currently away from the computer or console.
 IsBusy	Gets whether this friend is currently busy.
 IsDisposed	(Inherited from Gamer .)
 IsJoinable	Gets whether this friend is currently in a public session that can be joined.
 IsOnline	Gets whether this friend is currently online.
 IsPlaying	Gets whether this friend is currently playing a game.
 Presence	Gets a title-defined presence string describing what this friend is currently doing.
 SignedInGamers	(Inherited from Gamer .)
 Tag	(Inherited from Gamer .)

Public Methods

Name	Description
 BeginGetProfile	(Inherited from Gamer .)
 EndGetProfile	(Inherited from Gamer .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetProfile	(Inherited from Gamer .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also








Reference

[FriendGamer Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

FriendGamer Methods

Public Methods

	Name	Description
	BeginGetProfile	(Inherited from Gamer.)
	EndGetProfile	(Inherited from Gamer.)
	Equals	(Inherited from Object.)
	GetHashCode	(Inherited from Object.)
	GetProfile	(Inherited from Gamer.)
	GetType	(Inherited from Object.)
	ReferenceEquals	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also


















Reference

[FriendGamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

FriendGamer Properties

Public Properties

Name	Description
 FriendRequestReceivedFrom	Gets whether the local gamer who requested the friends list has received a friend request from this gamer.
 FriendRequestSentTo	Gets whether the local gamer who requested the friends list has sent a friend request to this gamer.
 Gamertag	(Inherited from Gamer .)
 HasVoice	Gets whether this friend currently has voice capability.
 InviteAccepted	Gets whether this friend has accepted an invitation from the local gamer who requested the friends list.
 InviteReceivedFrom	Gets whether the local gamer who requested the friends list has received an invitation from this friend.
 InviteRejected	Gets whether this friend has rejected an invitation from the local gamer who requested the friends list.
 InviteSentTo	Gets whether the local gamer who requested the friends list has sent an invitation to this friend.
 IsAway	Gets whether this friend is currently away from the computer or console.
 IsBusy	Gets whether this friend is currently busy.
 IsDisposed	(Inherited from Gamer .)
 IsJoinable	Gets whether this friend is currently in a public session that can be joined.
 IsOnline	Gets whether this friend is currently online.
 IsPlaying	Gets whether this friend is currently playing a game.
 Presence	Gets a title-defined presence string describing what this friend is currently doing.
 SignedInGamers	(Inherited from Gamer .)
 Tag	(Inherited from Gamer .)

See Also

Reference

[FriendGamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

FriendGamer.FriendRequestReceivedFrom Property

Gets whether the local gamer who requested the friends list has received a friend request from this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool FriendRequestReceivedFrom { get; }
```

Property Value

true if the local gamer who requested the friends list has received a friend request from this gamer; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.FriendRequestSentTo Property

Gets whether the local gamer who requested the friends list has sent a friend request to this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool FriendRequestSentTo { get; }
```

Property Value

true if the local gamer who requested the friends list has sent a friend request to this gamer; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.HasVoice Property

Gets whether this friend currently has voice capability.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasVoice { get; }
```

Property Value

true if this friend currently has voice capability; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.InviteAccepted Property

Gets whether this friend has accepted an invitation from the local gamer who requested the friends list.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool InviteAccepted { get; }
```

Property Value

true if this friend has accepted an invitation from the local gamer who requested the friends list; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.InviteReceivedFrom Property

Gets whether the local gamer who requested the friends list has received an invitation from this friend.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool InviteReceivedFrom { get; }
```

Property Value

true if the local gamer who requested the friends list has received an invitation from this friend; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.InviteRejected Property

Gets whether this friend has rejected an invitation from the local gamer who requested the friends list.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool InviteRejected { get; }
```

Property Value

true if this friend has rejected an invitation from the local gamer who requested the friends list; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.InviteSentTo Property

Gets whether the local gamer who requested the friends list has sent an invitation to this friend.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool InviteSentTo { get; }
```

Property Value

true if the local gamer who requested the friends list has sent an invitation to this friend; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.IsAway Property

Gets whether this friend is currently away from the computer or console.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsAway { get; }
```

Property Value

true if this friend is currently away from the computer or console; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.IsBusy Property

Gets whether this friend is currently busy.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsBusy { get; }
```

Property Value

true if this friend is currently busy; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.IsJoinable Property

Gets whether this friend is currently in a public session that can be joined.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsJoinable { get; }
```

Property Value

true if this friend is currently in a public session that can be joined; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.IsOnline Property

Gets whether this friend is currently online.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsOnline { get; }
```

Property Value

true if this friend is currently online; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.IsPlaying Property

Gets whether this friend is currently playing a game.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPlaying { get; }
```

Property Value

true if this friend is currently playing a game; **false** otherwise.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

FriendGamer.Presence Property

Gets a title-defined presence string describing what this friend is currently doing.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Presence { get; }
```

Property Value

Title-defined presence string describing what this friend is currently doing.

See Also

Reference

[FriendGamer Class](#)

[FriendGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults Class

Describes a gamer's preferred settings.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class GameDefaults
```

RemarksWherever possible, games should default to the values specified in **GameDefaults**.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Members](#)













[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune






GameDefaults Members

The following tables list the members exposed by the GameDefaults type.



Public Properties

Name	Description
 AccelerateWithButtons	Gets whether the gamer prefers to use controller buttons to accelerate in racing games.
 AutoAim	Gets whether the title should automatically correct the gamer's aim.
 AutoCenter	Gets whether the title should automatically center the view when the gamer moves.
 BrakeWithButtons	Gets whether the gamer prefers to use controller buttons to brake in racing games.
 ControllerSensitivity	Gets the preferred controller sensitivity setting for this gamer.
 GameDifficulty	Gets the preferred difficulty setting for this gamer.
 InvertYAxis	Gets whether the gamer prefers to invert the y-axis input of the controller.
 ManualTransmission	Gets whether the gamer prefers to drive a manual shift transmission in racing games.
 MoveWithRightThumbStick	Gets whether the gamer prefers to move using the right thumbstick.
 PrimaryColor	Gets the preferred color for the player character.
 RacingCameraAngle	Gets the gamer's preferred camera angle for racing games.
 SecondaryColor	Gets a secondary color selection for the player character.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference






[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GameDefaults Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also













Reference

[GameDefaults Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GameDefaults Properties

Public Properties

	Name	Description
	AccelerateWithButtons	Gets whether the gamer prefers to use controller buttons to accelerate in racing games.
	AutoAim	Gets whether the title should automatically correct the gamer's aim.
	AutoCenter	Gets whether the title should automatically center the view when the gamer moves.
	BrakeWithButtons	Gets whether the gamer prefers to use controller buttons to brake in racing games.
	ControllerSensitivity	Gets the preferred controller sensitivity setting for this gamer.
	GameDifficulty	Gets the preferred difficulty setting for this gamer.
	InvertYAxis	Gets whether the gamer prefers to invert the y-axis input of the controller.
	ManualTransmission	Gets whether the gamer prefers to drive a manual shift transmission in racing games.
	MoveWithRightThumbStick	Gets whether the gamer prefers to move using the right thumbstick.
	PrimaryColor	Gets the preferred color for the player character.
	RacingCameraAngle	Gets the gamer's preferred camera angle for racing games.
	SecondaryColor	Gets a secondary color selection for the player character.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GameDefaults.AccelerateWithButtons Property

Gets whether the gamer prefers to use controller buttons to accelerate in racing games.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AccelerateWithButtons { get; }
```

Property Value

true if the gamer prefers to use controller buttons to accelerate in racing games; **false** otherwise. If **false**, use triggers for acceleration.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.AutoAim Property

Gets whether the title should automatically correct the gamer's aim.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AutoAim { get; }
```

Property Value

true if the title should automatically correct the gamer's aim; **false** otherwise.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.AutoCenter Property

Gets whether the title should automatically center the view when the gamer moves.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AutoCenter { get; }
```

Property Value

true if the title should automatically center the view when the gamer moves; **false** otherwise.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.BrakeWithButtons Property

Gets whether the gamer prefers to use controller buttons to brake in racing games.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool BrakeWithButtons { get; }
```

Property Value

true if the gamer prefers to use controller buttons to brake in racing games; **false** otherwise. If **false**, use the triggers for braking.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.ControllerSensitivity Property

Gets the preferred controller sensitivity setting for this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ControllerSensitivity ControllerSensitivity { get; }
```

Property Value

The preferred controller sensitivity setting for this gamer.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.GameDifficulty Property

Gets the preferred difficulty setting for this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GameDifficulty GameDifficulty { get; }
```

Property Value

The preferred difficulty setting for this gamer.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.InvertYAxis Property

Gets whether the gamer prefers to invert the y-axis input of the controller.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool InvertYAxis { get; }
```

Property Value

true if the gamer prefers to invert the y-axis input of the controller; **false** otherwise.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.ManualTransmission Property

Gets whether the gamer prefers to drive a manual shift transmission in racing games.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool ManualTransmission { get; }
```

Property Value

true if the gamer prefers to drive a manual shift transmission in racing games; **false** otherwise.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.MoveWithRightThumbStick Property

Gets whether the gamer prefers to move using the right thumbstick.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveWithRightThumbStick { get; }
```

Property Value

true if the gamer prefers to move using the right thumbstick; **false** otherwise. If this value is **false**, use the left thumbstick for player movement.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.PrimaryColor Property

Gets the preferred color for the player character.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<Color> PrimaryColor { get; }
```

Property Value

The preferred color for the player character.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.RacingCameraAngle Property

Gets the gamer's preferred camera angle for racing games.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RacingCameraAngle RacingCameraAngle { get; }
```

Property Value

The gamer's preferred camera angle for racing games.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDefaults.SecondaryColor Property

Gets a secondary color selection for the player character.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<Color> SecondaryColor { get; }
```

Property Value

A secondary color selection for the player character.

See Also

Reference

[SignedInGamer.GameDefaults Property](#)

[GameDefaults Class](#)

[GameDefaults Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameDifficulty Enumeration

Indicates how difficult this gamer likes things to be.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GameDifficulty
```

Members

Member name	Description
Easy	Below-average difficulty.
Hard	Above-average difficulty.
Normal	Average difficulty.

See Also

Reference

[GameDefaults.GameDifficulty Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer Class

Abstract base class for types that represent game players (profiles that have an associated gamertag). The concrete types [SignedInGamer](#) and [NetworkGamer](#) derive from this.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class Gamer
```

See Also

Reference

[Gamer Members](#)






[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









Gamer Members

The following tables list the members exposed by the Gamer type.



Public Properties

	Name	Description
	Gamertag	Gets the gamertag string.
	IsDisposed	Gets a value indicating whether the object is disposed.
 	SignedInGamers	Represents a collection of all gamers on the local system.
	Tag	Gets or sets a custom object which can be used to attach arbitrary user defined data to the gamer.

Public Methods

	Name	Description
	BeginGetProfile	Starts an asynchronous profile read operation.
	EndGetProfile	Ends an asynchronous profile read operation.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetProfile	Reads profile data for this gamer.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[Gamer Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

Gamer Methods

Public Methods

	Name	Description
	BeginGetProfile	Starts an asynchronous profile read operation.
	EndGetProfile	Ends an asynchronous profile read operation.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetProfile	Reads profile data for this gamer.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Gamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Gamer.BeginGetProfile Method

Starts an asynchronous profile read operation.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IAsyncResult BeginGetProfile (  
    AsyncCallback callback,  
    Object asyncState  
)
```

Parameters

callback

The method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method. Call [EndGetProfile](#) to access these results.

Exceptions

Exception type	Condition
ObjectDisposedException	This object has already been disposed.

See Also

Reference

[GetProfile](#)

[EndGetProfile](#)

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer.EndGetProfile Method

Ends an asynchronous profile read operation.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerProfile EndGetProfile (  
    IAsyncResult result  
)
```

Parameters

result

An [IAsyncResult](#) used to track the progress of the operation.

Return Value

The profile data for this gamer.

See Also

Reference

[GetProfile](#)

[BeginGetProfile](#)

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer.GetProfile Method

Reads profile data for this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerProfile GetProfile ()
```

Return Value

The profile data for this gamer.

Remarks This operation can take some time if called on a remote gamer instance, in which case you might prefer to use the non-blocking alternative [BeginGetProfile](#), but will complete quickly when used with a locally signed in profile.

See Also

Reference

[BeginGetProfile](#)

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[Gamer Class](#)





[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer Properties

Public Properties

	Name	Description
	Gamertag	Gets the gamertag string.
	IsDisposed	Gets a value indicating whether the object is disposed.
 S	SignedInGamers	Represents a collection of all gamers on the local system.
	Tag	Gets or sets a custom object which can be used to attach arbitrary user defined data to the gamer.

See Also

Reference

[Gamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Gamer.Gamertag Property

Gets the gamertag string.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Gamertag { get; }
```

Property Value

Name of the gamertag.

See Also

Reference

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer.IsDisposed Property

Gets a value indicating whether the object is disposed.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer.SignedInGamers Property

Represents a collection of all gamers on the local system. The collection is sorted by the gamer index, skipping non-existent gamer indices. For example, if gamers 1 and 3 are signed in, the collection contains two instances of [SignedInGamer](#), with gamer 1 first in the collection.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static SignedInGamerCollection SignedInGamers { get; }
```

Property Value

Collection of current gamers on the local system.

See Also

Reference

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Gamer.Tag Property

Gets or sets a custom object which can be used to attach arbitrary user defined data to the gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

A custom object used to attach arbitrary user defined data to the gamer.

See Also

Reference

[Gamer Class](#)

[Gamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerCollection Generic Class

Represents a collection of gamers. This collection cannot be modified and is updated automatically during the call to [Update](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GamerCollection<T> : ReadOnlyCollection<T>, IEnumerable<Gamer>, IEnumerable where T : Gamer
```

Remarks

Access this collection using the following methods.

- [AllGamers](#)
- [LocalGamers](#)
- [RemoteGamers](#)

See Also

Reference

[GamerCollection Members](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


GamerCollection Members

The following tables list the members exposed by the GamerCollection type.

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)


Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	Returns a strongly typed GamerCollection.GamerCollectionEnumerator Generic Structure that can iterate through a GamerCollection .
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

Explicit Interface Implementations

	Name	Description
	System.Collections.Generic.IEnumerable<Microsoft.Xna.Framework.GamerServices.Gamer>.GetEnumerator	Returns an enumerator that can iterate through the collection.

See Also

Reference

[GamerCollection Generic Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerCollection Methods

Public Methods

Name	Description
 Contains	(Inherited from ReadOnlyCollection .)
 CopyTo	(Inherited from ReadOnlyCollection .)
 Equals	(Inherited from Object .)
 GetEnumerator	Returns a strongly typed GamerCollection.GamerCollectionEnumerator Generic Structure that can iterate through a GamerCollection .
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from ReadOnlyCollection .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.IEnumerable<Microsoft.Xna.Framework.GamerServices.Gamer>.GetEnumerator	Returns an enumerator that can iterate through the collection.

See Also

Reference

[GamerCollection Generic Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerCollection.GetEnumerator Method

Returns a strongly typed [GamerCollection.GamerCollectionEnumerator Generic Structure](#) that can iterate through a [GamerCollection](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerCollectionEnumerator GetEnumerator ()
```

Return Value

The [GamerCollection.GamerCollectionEnumerator Generic Structure](#).

See Also

Reference

[GamerCollection Generic Class](#)

[GamerCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.IEnumerable<Microsoft.Xna.Framework.GamerServices.Gamer>.GetEnumerator Method

Returns an enumerator that can iterate through the collection.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator<Gamer> System.Collections.Generic.IEnumerable<Microsoft.Xna.Framework.GamerServices.Gamer>.GetEnumerator ()
```

Return Value

Enumerator that can iterate through the collection.

See Also

Reference

[GamerCollection Generic Class](#)



[GamerCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[GamerCollection](#) Generic Class

[Microsoft.Xna.Framework.GamerServices](#) Namespace

GamerCollection.GamerCollectionEnumerator Generic Structure

Provides the ability to iterate through the gamers in an [GamerCollection](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamerCollection.GamerCollectionEnumerator<T> : IEnumerable<T>, IDisposable, I  
Enumerator where T : Gamer
```

See Also

Reference

[GamerCollection.GamerCollectionEnumerator Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GamerCollection.GamerCollectionEnumerator Members

The following tables list the members exposed by the GamerCollection.GamerCollectionEnumerator type.



Public Properties

	Name	Description
	Current	Gets the current element in the GamerCollection .



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the GamerCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the GamerCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the GamerCollection .

See Also






Reference

[GamerCollection.GamerCollectionEnumerator Generic Structure](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerCollection.GamerCollectionEnumerator Methods



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the GamerCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the GamerCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the GamerCollection .

See Also

Reference

[GamerCollection.GamerCollectionEnumerator Generic Structure](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerCollection.GamerCollectionEnumerator.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[GamerCollection.GamerCollectionEnumerator Generic Structure](#)

[GamerCollection.GamerCollectionEnumerator Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerCollection.GamerCollectionEnumerator.MoveNext Method

Advances the enumerator to the next element of the [GamerCollection](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveNext ()
```

Return Value

true if the enumerator was successfully advanced to the next element; **false** if the enumerator has passed the end of the collection.

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

After an enumerator is created or after a call to the [System.Collections.IEnumerator.Reset Method](#) method, an enumerator is positioned before the first element of the collection, and the first call to the **MoveNext** method moves the enumerator over the first element of the collection.

After the end of the collection is passed, subsequent calls to **MoveNext** return false until [System.Collections.IEnumerator.Reset Method](#) is called.

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to **MoveNext** or [System.Collections.IEnumerator.Reset Method](#) throws an [InvalidOperationException](#).

See Also

Reference

[GamerCollection.GamerCollectionEnumerator Generic Structure](#)

[GamerCollection.GamerCollectionEnumerator Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerator.Reset Method

Sets the enumerator to its initial position, which is before the first element in the [GamerCollection](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.IEnumerator.Reset ()
```

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

An enumerator remains valid as long as the collection remains unchanged. If you make changes to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated, and the next call to the [MoveNext](#) method or the **System.Collections.IEnumerator.Reset** method throws an [InvalidOperationException](#).

Note

Notes to Implementers: All calls to **System.Collections.IEnumerator.Reset** must result in the same state for the enumerator. The preferred implementation is to move the enumerator to the beginning of the collection, before the first element. This invalidates the enumerator if the collection has been modified since the enumerator was created, which is consistent with [MoveNext](#) and [Current](#).

See Also

Reference

[GamerCollection.GamerCollectionEnumerator Generic Structure](#)


[GamerCollection.GamerCollectionEnumerator Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerCollection.GamerCollectionEnumerator Properties

Public Properties

	Name	Description
	Current	Gets the current element in the GamerCollection .

See Also

Reference

[GamerCollection.GamerCollectionEnumerator](#) Generic Structure

[Microsoft.Xna.Framework.GamerServices](#) Namespace

GamerCollection.GamerCollectionEnumerator.Current Property

Gets the current element in the [GamerCollection](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public T Current { get; }
```

Property Value

The current element in the [GamerCollection](#).

See Also

Reference

[GamerCollection.GamerCollectionEnumerator Generic Structure](#)

[GamerCollection.GamerCollectionEnumerator Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerCollection.GamerCollectionEnumerator.System.Collections.IEnumerator.Current Property

Gets the current element in the [GamerCollection](#) as a [Object](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Object System.Collections.IEnumerator.Current { get; }
```

Property Value

The current element in the [GamerCollection](#) as a [Object](#).

See Also

Reference

[GamerCollection.GamerCollectionEnumerator](#) Generic Structure

[GamerCollection.GamerCollectionEnumerator](#) Members

[Microsoft.Xna.Framework.GamerServices](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPresence Class

Provides properties to set the rich presence state for a locally signed-in gamer profile.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class GamerPresence
```

Remarks

Zune Specific Information

Setting presence information is not supported on the Zune platform. On the Zune platform, setting presence information for a signed-in gamer will not change the presence information on Xbox LIVE.

See Also

Tasks

[How To: Add Presence Information](#)

Reference

[SignedInGamer.Presence Property](#)

[Gamer.SignedInGamers Property](#)

[GamerPresence Members](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GamerPresence Members

The following tables list the members exposed by the GamerPresence type.



Public Properties

	Name	Description
	PresenceMode	Gets or sets the current presence mode for this gamer.
	PresenceValue	Gets or sets a custom presence value.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Add Presence Information](#)

Reference

[SignedInGamer.Presence Property](#)






[Gamer.SignedInGamers Property](#)

[GamerPresence Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPresence Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Add Presence Information](#)

Reference

[SignedInGamer.Presence Property](#)



[Gamer.SignedInGamers Property](#)

[GamerPresence Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPresence Properties

Public Properties

	Name	Description
	PresenceMode	Gets or sets the current presence mode for this gamer.
	PresenceValue	Gets or sets a custom presence value.

See Also

Tasks

[How To: Add Presence Information](#)

Reference

[SignedInGamer.Presence Property](#)

[Gamer.SignedInGamers Property](#)

[GamerPresence Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPresence.PresenceMode Property

Gets or sets the current presence mode for this gamer. You will see this status string when you view a friend through the Guide or on Xbox.com.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPresenceMode PresenceMode { get; set; }
```

Property Value

The current presence mode for this gamer.

Remarks

Zune Specific Information

Setting presence information is not supported on the Zune platform. On the Zune platform, setting presence information for a signed-in gamer will not change the presence information on Xbox LIVE.

Some of the presence mode options include a number that can be set using the [PresenceValue](#) property:

- [GamerPresenceMode.Stage](#)
- [GamerPresenceMode.Level](#)
- [GamerPresenceMode.Score](#)
- [GamerPresenceMode.CoopStage](#)
- [GamerPresenceMode.CoopLevel](#)
- [GamerPresenceMode.VersusScore](#)

Note

A game title will not be displayed in a player's presence information on the Xbox LIVE service until that game passes peer review. Once a game passes peer review, the presence information for a player shows three things: a confirmation the player is playing an Xbox LIVE Indie Game, the game's title, and the presence information set by [SignedInGamer.Presence](#). For example:

```
Xbox LIVE Indie Game
My Game Title
Fighting the Boss
```

While a game is in development, but before the game passes peer review, the presence information indicates the developer is using the XNA Creators Club, as shown:

```
XNA Creators Club
creators.xna.com
Fighting the Boss
```

See Also

Tasks

[How To: Add Presence Information](#)

Reference

[GamerPresence.PresenceValue Property](#)

[SignedInGamer.Presence Property](#)

[Gamer.SignedInGamers Property](#)

[GamerPresence Class](#)

[GamerPresence Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPresence.PresenceValue Property

Gets or sets a custom presence value. You can insert this into the presence string if you select one of the presence modes that includes a value. For example, you might select [GamerPresenceMode.Score](#) or [GamerPresenceMode.Level](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PresenceValue { get; set; }
```

Property Value

A custom presence value, which indicates a stage, score, or level to display as a part of the presence string.

Remarks

Zune Specific Information

Setting presence information is not supported on the Zune platform. On the Zune platform, setting presence information for a signed-in gamer will not change the presence information on Xbox LIVE.

The following [GamerPresenceMode](#) settings, specified in [PresenceMode](#), may include a numeric value as specified by [PresenceValue](#):

- [GamerPresenceMode.Stage](#)
- [GamerPresenceMode.Level](#)
- [GamerPresenceMode.Score](#)
- [GamerPresenceMode.CoopStage](#)
- [GamerPresenceMode.CoopLevel](#)
- [GamerPresenceMode.VersusScore](#)

See Also

Reference

[GamerPresence.PresenceMode Property](#)

[SignedInGamer.Presence Property](#)

[Gamer.SignedInGamers Property](#)

[GamerPresence Class](#)

[GamerPresence Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPresenceMode Enumeration

Settings defining the status string that will appear when you view a friend through the Xbox LIVE Guide or on Xbox.com. Use the [PresenceMode](#) property to set this option.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GamerPresenceMode
```

Members

Member name	Description
ArcadeMode	Arcade Mode
AtMenu	At Menu
BattlingBoss	Battling Boss
CampaignMode	Campaign Mode
ChallengeMode	Challenge Mode
ConfiguringSettings	Configuring Settings
CoOpLevel	Co-Op: Level. Includes a numeric value specified with PresenceValue .
CoOpStage	Co-Op: Stage. Includes a numeric value specified with PresenceValue .
CornflowerBlue	Cornflower Blue
CustomizingPlayer	Customizing Player
DifficultyEasy	Difficulty: Easy
DifficultyExtreme	Difficulty: Extreme
DifficultyHard	Difficulty: Hard
DifficultyMedium	Difficulty: Medium
EditingLevel	Editing Level
ExplorationMode	Exploration Mode
FoundSecret	Found Secret
FreePlay	Free Play
GameOver	Game Over
InCombat	In Combat
InGameStore	In Game Store
Level	Level. Includes a numeric value specified with PresenceValue .
LocalCoOp	Local Co-Op
LocalVersus	Local Versus
LookingForGames	Looking For Games
Losing	Losing
Multiplayer	Multiplayer
NearlyFinished	Nearly Finished
None	No Presence String Displayed
OnARoll	On a Roll
OnlineCoOp	Online Co-Op
OnlineVersus	Online Versus
Outnumbered	Outnumbered
Paused	Paused
PlayingMinigame	Playing Minigame
PlayingWithFriends	Playing With Friends
PracticeMode	Practice Mode
PuzzleMode	Puzzle Mode
ScenarioMode	Scenario Mode
Score	Score. Includes a numeric value specified with PresenceValue .

ScoresTied	Score is Tied
SettingUpMatch	Setting Up Match
SinglePlayer	Single Player
Stage	Stage. Includes a numeric value specified with PresenceValue .
StartingGame	Starting Game
StoryMode	Story Mode
StuckOnAHardBit	Stuck on a Hard Bit
SurvivalMode	Survival Mode
TimeAttack	Time Attack
TryingForRecord	Trying For Record
TutorialMode	Tutorial Mode
VersusComputer	Versus Computer
VersusScore	Versus: Score. Includes a numeric value specified with PresenceValue .
WaitingForPlayers	Waiting For Players
WaitingInLobby	Waiting In Lobby
WastingTime	Wasting Time
WatchingCredits	Watching Credits
WatchingCutscene	Watching Cutscene
Winning	Winning
WonTheGame	Won the Game

Remarks

<p> Zune Specific Information</p> <p>Setting presence information is not supported on the Zune platform. On the Zune platform, setting presence information for a signed-in gamer will not change the presence information on Xbox LIVE.</p>

The following [GamerPresenceMode](#) settings, specified in [PresenceMode](#), may include a numeric value as specified by [PresenceValue](#):

- [GamerPresenceMode.Stage](#)
- [GamerPresenceMode.Level](#)
- [GamerPresenceMode.Score](#)
- [GamerPresenceMode.CoopStage](#)
- [GamerPresenceMode.CoopLevel](#)
- [GamerPresenceMode.VersusScore](#)

See Also

Reference

[GamerPresence.PresenceMode Property](#)

[GamerPresence.PresenceValue Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivilegeException Class

Thrown if a [gamer services](#) or [multiplayer](#) API is called without a valid, signed-in profile.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]
public class GamerPrivilegeException : Exception
```

Remarks

This exception is thrown if any of the following conditons are met.

- No profile is currently signed in.
- The signed-in profile has parental control settings prohibiting the requested action.

See Also

Reference

[GamerPrivilegeException Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GamerPrivilegeException Members

The following tables list the members exposed by the GamerPrivilegeException type.


Public Constructors

Name	Description
 GamerPrivilegeException	Overloaded. Creates an empty instance of GamerPrivilegeException.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamerPrivilegeException Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivilegeException Constructor

Creates an empty instance of **GamerPrivilegeException**.

Overload List

Name	Description
GamerPrivilegeException ()	Creates an instance of GamerPrivilegeException .
GamerPrivilegeException (SerializationInfo, StreamingContext)	Initializes a new instance of GamerPrivilegeException with the specified streaming context.
GamerPrivilegeException (String)	Initializes a new instance of GamerPrivilegeException with the specified error message.
GamerPrivilegeException (String, Exception)	Initializes a new instance of GamerPrivilegeException with the specified error message and inner exception.

See Also

Reference

[GamerPrivilegeException Class](#)

[GamerPrivilegeException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivilegeException Constructor ()

Creates an instance of **GamerPrivilegeException**.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivilegeException ()
```

See Also

Reference

[GamerPrivilegeException Class](#)

[GamerPrivilegeException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivilegeException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **GamerPrivilegeException** with the specified streaming context.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected GamerPrivilegeException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Describes the gamer services or multiplayer API being accessed when the exception occurred.

context

Describes the stream where the exception occurred.

See Also

Reference

[GamerPrivilegeException Class](#)

[GamerPrivilegeException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GamerPrivilegeException Constructor (String)

Initializes a new instance of **GamerPrivilegeException** with the specified error message.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivilegeException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[GamerPrivilegeException Class](#)

[GamerPrivilegeException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivilegeException Constructor (String, Exception)

Initializes a new instance of **GamerPrivilegeException** with the specified error message and inner exception.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivilegeException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The inner exception related to this exception.

See Also

Reference

[GamerPrivilegeException Class](#)







[GamerPrivilegeException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivilegeException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[GamerPrivilegeException Class](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivilegeException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[GamerPrivilegeException Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivileges Class

Describes what operations a gamer is allowed to perform.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class GamerPrivileges
```

Remarks

GamerPrivileges can be defined by parental control settings, and will also be set automatically in response to things like age, region, and whether the gamer has a LIVE Gold or LIVE Silver account. Games do not need to explicitly check privileges, because a [GamerPrivilegeException](#) will be thrown if they try to perform an unsupported operation, but these privilege bits may be useful to detect that an operation is unavailable before calling it so as to grey-out the relevant menu option.

See Also

Reference

[SignedInGamer.Privileges Property](#)

[GamerPrivileges Members](#)







[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GamerPrivileges Members

The following tables list the members exposed by the GamerPrivileges type.



Public Properties

Name	Description
 AllowCommunication	Checks whether this gamer is allowed to send and receive communications using voice, text, messaging, or game invites.
 AllowOnlineSessions	Checks whether this gamer is allowed to play in online multiplayer sessions.
 AllowProfileViewing	Checks whether this gamer is allowed to view the profiles of other gamers.
 AllowPurchaseContent	Checks whether this gamer is allowed to purchase content from LIVE Marketplace.
 AllowTradeContent	Checks whether this gamer is allowed to trade content with other gamers.
 AllowUserCreatedContent	Checks whether this gamer is allowed to access user content that was created by other gamers.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[GamerPrivileges Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivileges Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[GamerPrivileges Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivileges Properties

Public Properties

	Name	Description
	AllowCommunication	Checks whether this gamer is allowed to send and receive communications using voice, text, messaging, or game invites.
	AllowOnlineSessions	Checks whether this gamer is allowed to play in online multiplayer sessions.
	AllowProfileViewing	Checks whether this gamer is allowed to view the profiles of other gamers.
	AllowPurchaseContent	Checks whether this gamer is allowed to purchase content from LIVE Marketplace.
	AllowTradeContent	Checks whether this gamer is allowed to trade content with other gamers.
	AllowUserCreatedContent	Checks whether this gamer is allowed to access user content that was created by other gamers.

See Also

Reference

[GamerPrivileges Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerPrivileges.AllowCommunication Property

Checks whether this gamer is allowed to send and receive communications using voice, text, messaging, or game invites.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivilegeSetting AllowCommunication { get; }
```

Property Value

true if this gamer is allowed to send and receive communications using voice, text, messaging, or game invites; **false** otherwise.

Remarks

Titles do not need to do anything special to disable voice if this privilege is disabled. The framework will handle that automatically.

See Also

Reference

[SignedInGamer.Privileges Property](#)

[Guide.BeginShowMessageBox Method](#)

[Guide.ShowComposeMessage Method](#)

[Guide.ShowGameInvite Method](#)

[LocalNetworkGamer.EnableSendVoice Method](#)

[GamerPrivileges Class](#)

[GamerPrivileges Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivileges.AllowOnlineSessions Property

Checks whether this gamer is allowed to play in online multiplayer sessions.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AllowOnlineSessions { get; }
```

Property Value

true if this gamer is allowed to play in online multiplayer sessions; **false** otherwise.

Remarks

This may be disabled based on parental control settings. Also, it is disabled if the gamer does not have a Xbox LIVE Gold or LIVE Silver membership. This property does not affect system link sessions, which are always available.

While in limited trial mode, this property will return **false**, which means gamers won't be able to use the Xbox LIVE servers to connect to other machines over the Internet. It is possible that the game may change from trial mode at any time, including during actual gameplay. For more information, please see [Guide.IsTrialMode](#).

See Also

Reference

[Guide.IsTrialMode Property](#)

[SignedInGamer.Privileges Property](#)

[GamerPrivileges Class](#)

[GamerPrivileges Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivileges.AllowProfileViewing Property

Checks whether this gamer is allowed to view the profiles of other gamers.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivilegeSetting AllowProfileViewing { get; }
```

Property Value

true if this gamer is allowed to view the profiles of other gamers; **false** otherwise.

See Also

Reference

[SignedInGamer.Privileges Property](#)

[Guide.ShowGamerCard Method](#)

[GamerPrivileges Class](#)

[GamerPrivileges Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivileges.AllowPurchaseContent Property

Checks whether this gamer is allowed to purchase content from LIVE Marketplace.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AllowPurchaseContent { get; }
```

Property Value

true if this gamer is allowed to purchase content from LIVE Marketplace; **false** otherwise.

See Also

Reference

[Guide.ShowMarketplace Method](#)

[SignedInGamer.Privileges Property](#)

[GamerPrivileges Class](#)

[GamerPrivileges Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivileges.AllowTradeContent Property

Checks whether this gamer is allowed to trade content with other gamers.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AllowTradeContent { get; }
```

Property Value

true if this gamer is allowed to trade content with other gamers; **false** otherwise.

See Also

Reference

[SignedInGamer.Privileges Property](#)

[GamerPrivileges Class](#)

[GamerPrivileges Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivileges.AllowUserCreatedContent Property

Checks whether this gamer is allowed to access user content that was created by other gamers.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivilegeSetting AllowUserCreatedContent { get; }
```

Property Value

true if this gamer is allowed to access user content that was created by other gamers; **false** otherwise.

See Also

Reference

[SignedInGamer.Privileges Property](#)

[GamerPrivileges Class](#)

[GamerPrivileges Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerPrivilegeSetting Enumeration

Describes the conditions in which a privilege is available.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GamerPrivilegeSetting
```

Members

Member name	Description
Blocked	This privilege is not available for the current gamer profile.
Everyone	This privilege is available for the current gamer profile.
FriendsOnly	This privilege is only available for friends of the current gamer profile. Use the IsFriend method to check which gamers are friends.

See Also

Reference

[GamerPrivileges.AllowProfileViewing Property](#)

[GamerPrivileges.AllowCommunication Property](#)

[GamerPrivileges.AllowUserCreatedContent Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile Class

Profile settings describing information about a gamer such as the gamer's motto, reputation, and gamer picture. This data is accessible for both locally signed in profiles and remote gamers that you are playing with in a multiplayer session.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class GamerProfile : IDisposable
```

See Also

Reference

[GamerProfile Members](#)










[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







GamerProfile Members

The following tables list the members exposed by the GamerProfile type.



Public Properties

Name	Description
 GamerPicture	Gets the gamer picture, which will be 64 pixels by 64 pixels.
 GamerScore	Gets the GamerScore of this gamer.
 GamerZone	Gets the GamerZone setting.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Motto	Gets the gamer motto string.
 Region	Gets the region of this gamer.
 Reputation	Gets the gamer reputation, as a number of stars ranging 0 to 5.
 TitlesPlayed	Gets the number of titles this gamer has played.
 TotalAchievements	Gets the total number of achievements this gamer has obtained.

Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Use Gamertags and Gamer Pictures](#)

Reference







[Gamer.SignedInGamers Property](#)

[GamerProfile Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerProfile Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamerProfile Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerProfile.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[GamerProfile Class](#)










[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile Properties

Public Properties

	Name	Description
	GamerPicture	Gets the gamer picture, which will be 64 pixels by 64 pixels.
	GamerScore	Gets the GamerScore of this gamer.
	GamerZone	Gets the GamerZone setting.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Motto	Gets the gamer motto string.
	Region	Gets the region of this gamer.
	Reputation	Gets the gamer reputation, as a number of stars ranging 0 to 5.
	TitlesPlayed	Gets the number of titles this gamer has played.
	TotalAchievements	Gets the total number of achievements this gamer has obtained.

See Also

Reference

[GamerProfile Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerProfile.GamerPicture Property

Gets the gamer picture, which will be 64 pixels by 64 pixels.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GamerPicture { get; }
```

Property Value

The gamer picture, which will be 64 pixels by 64 pixels.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.GamerScore Property

Gets the GamerScore of this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int GamerScore { get; }
```

Property Value

The GamerScore of this gamer.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.GamerZone Property

Gets the [GamerZone](#) setting.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerZone GamerZone { get; }
```

Property Value

The [GamerZone](#) setting.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.Motto Property

Gets the gamer motto string.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Motto { get; }
```

Property Value

The gamer motto.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.Region Property

Gets the region of this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RegionInfo Region { get; }
```

Property Value

The region of this gamer.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.Reputation Property

Gets the gamer reputation, as a number of stars ranging 0 to 5.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Reputation { get; }
```

Property Value

The gamer reputation, representing a number of stars ranging 0 to 5.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.TitlesPlayed Property

Gets the number of titles this gamer has played.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int TitlesPlayed { get; }
```

Property Value

The number of titles this gamer has played.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerProfile.TotalAchievements Property

Gets the total number of achievements this gamer has obtained.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int TotalAchievements { get; }
```

Property Value

The total number of achievements this gamer has obtained.

See Also

Reference

[GamerProfile Class](#)

[GamerProfile Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesComponent Class

Wraps the functionality of the [GamerServicesDispatcher](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GamerServicesComponent : GameComponent
```

Remarks If a game will use gamer services, a **GamerServicesComponent** should be added to the game's [Components](#) collection. **GamerServicesComponent** will take care of initializing and updating the [GamerServicesDispatcher](#) at the appropriate times.

See Also

Reference

[GamerServicesComponent Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




GamerServicesComponent Members

The following tables list the members exposed by the GamerServicesComponent type.









Public Constructors

Name	Description
 GamerServicesComponent	Creates a new GamerServicesComponent .




Public Properties

Name	Description
 Enabled	(Inherited from GameComponent .)
 Game	(Inherited from GameComponent .)
 UpdateOrder	(Inherited from GameComponent .)




Public Methods

Name	Description
 Dispose	(Inherited from GameComponent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Initialize	Initializes the GamerServicesDispatcher .
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Update	Updates the GamerServicesDispatcher .

Protected Methods

Name	Description
 MemberwiseClone	(Inherited from Object .)
 OnEnabledChanged	(Inherited from GameComponent .)
 OnUpdateOrderChanged	(Inherited from GameComponent .)

Public Events

Name	Description
 Disposed	(Inherited from GameComponent .)
 EnabledChanged	(Inherited from GameComponent .)
 UpdateOrderChanged	(Inherited from GameComponent .)

See Also

Reference

[GamerServicesComponent Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesComponent Constructor

Creates a new [GamerServicesComponent](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerServicesComponent (
    Game game
)
```

Parameters

game

The game that will be associated with this component.

Exceptions

Exception type	Condition
GamerServicesNotAvailableException	<ul style="list-style-type: none"> If the current game uses the XNA Framework Redistributable. This file does not include Games for Windows – LIVE functionality. If another Games for Windows – LIVE title is running already.
ArgumentNullException	<i>serviceProvider</i> is null .
InvalidOperationException	The gamer services functionality is already initialized.

Remarks

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

See Also

Reference

[GamerServicesComponent Class](#)









[GamerServicesComponent Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesComponent Methods

Public Methods

	Name	Description
	Dispose	(Inherited from GameComponent .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Initialize	Initializes the GamerServicesDispatcher .
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Update	Updates the GamerServicesDispatcher .

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	OnEnabledChanged	(Inherited from GameComponent .)
	OnUpdateOrderChanged	(Inherited from GameComponent .)

See Also

Reference

[GamerServicesComponent Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesComponent.Initialize Method

Initializes the [GamerServicesDispatcher](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override void Initialize ()
```

Exceptions

Exception type	Condition
GamerServicesNotAvailableException	<ul style="list-style-type: none"> If the current game uses the XNA Framework Redistributable. This file does not include Games for Windows – LIVE functionality. If another Games for Windows – LIVE title is running already.
ArgumentNullException	<i>serviceProvider</i> is null .
InvalidOperationException	The gamer services functionality is already initialized.

Remarks

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

See Also

Reference

[GamerServicesComponent Class](#)

[GamerServicesComponent Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesComponent.Update Method

Updates the [GamerServicesDispatcher](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override void Update (  
    gameTime gameTime  
)
```

Parameters

gameTime

The game timing state.

See Also

Reference

[GamerServicesComponent Class](#)




[GamerServicesComponent Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesComponent Properties

Public Properties

	Name	Description
	Enabled	(Inherited from GameComponent .)
	Game	(Inherited from GameComponent .)
	UpdateOrder	(Inherited from GameComponent .)

See Also




Reference

[GamerServicesComponent Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesComponent Events

Public Events

	Name	Description
	Disposed	(Inherited from GameComponent .)
	EnabledChanged	(Inherited from GameComponent .)
	UpdateOrderChanged	(Inherited from GameComponent .)

See Also

Reference

[GamerServicesComponent Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesDispatcher Class

Implements the Windows-specific portion of a **GamerServicesDispatcher** class.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class GamerServicesDispatcher
```

Remarks

Use the [GamerServicesDispatcher](#) class to access gamer services functionality from your game. This functionality includes, among other services, displaying the guide from your game, determining which profiles are checked in, and determining the privileges for a specific gamer profile.

During startup of your game, initialize an instance of **GamerServicesDispatcher**. During the lifetime of your game, call [Update](#) for every frame.

Commonly, you do not directly interact with the **GamerServicesDispatcher** object. Instead, use the [GamerServicesComponent](#) class, which wraps the functionality of **GamerServicesDispatcher**. However, for applications that do not use the default game template, you must interact with the **GamerServicesDispatcher** instance of your application.

See Also

Tasks

[How To: Initialize and Update the Gamer Services Dispatcher](#)

Concepts

[Gamer Services Overview](#)

Reference

[GamerServicesDispatcher Members](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










GamerServicesDispatcher Members

The following tables list the members exposed by the GamerServicesDispatcher type.



Public Properties

	Name	Description
	IsInitialized	Determines if Initialize has been called.
	WindowHandle	Gets or sets the handle to the underlying game window.



Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	 Initialize	Initializes gamer services functionality for the game.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	 Update	Updates the status of gamer services and raises related events.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Public Events

	Name	Description
	 InstallingTitleUpdate	Notifies the game when a Games for Windows - LIVE title update is being installed.

See Also

Tasks

[How To: Initialize and Update the Gamer Services Dispatcher](#)

Concepts

[Gamer Services Overview](#)










Reference

[GamerServicesDispatcher Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesDispatcher Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
 	Initialize	Initializes gamer services functionality for the game.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
 	Update	Updates the status of gamer services and raises related events.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamerServicesDispatcher Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesDispatcher.Initialize Method

Initializes gamer services functionality for the game.

Call this method during startup of your game.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Initialize (
    IServiceProvider serviceProvider
)
```

Parameters

serviceProvider

The service provider to be used when locating additional services services. For example, [IGraphicsDeviceService](#).

Exceptions

Exception type	Condition
GamerServicesNotAvailableException	<ul style="list-style-type: none"> If the current game uses the XNA Framework Redistributable. This file does not include Games for Windows – LIVE functionality. If another Games for Windows – LIVE title is running already.
ArgumentNullException	<i>serviceProvider</i> is null .
InvalidOperationException	The gamer services functionality is already initialized.

Remarks

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

See Also

Tasks

[How To: Initialize and Update the Gamer Services Dispatcher](#)

Concepts

[Gamer Services Overview](#)

Reference

[GamerServicesDispatcher Class](#)

[GamerServicesDispatcher Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesDispatcher.Update Method

Updates the status of gamer services and raises related events. It is recommended that you call this method once per frame.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Update ()
```

See Also

Tasks

[How To: Initialize and Update the Gamer Services Dispatcher](#)

Concepts

[Gamer Services Overview](#)

Reference

[GamerServicesDispatcher Class](#)



[GamerServicesDispatcher Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesDispatcher Properties

Public Properties

	Name	Description
 S	IsInitialized	Determines if Initialize has been called.
 S	WindowHandle	Gets or sets the handle to the underlying game window.

See Also

Reference

[GamerServicesDispatcher Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesDispatcher.IsInitialized Property

Determines if [Initialize](#) has been called.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsInitialized { get; }
```

Property Value

true if gamer services have been initialized; **false** otherwise.

See Also

Reference

[GamerServicesDispatcher Class](#)

[GamerServicesDispatcher Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesDispatcher.WindowHandle Property

Gets or sets the handle to the underlying game window.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IntPtr WindowHandle { get; set; }
```

Property Value

Handle to a window.

See Also

Reference

[GamerServicesDispatcher Class](#)


[GamerServicesDispatcher Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesDispatcher Events

Public Events

	Name	Description
 S	InstallingTitleUpdate	Notifies the game when a Games for Windows - LIVE title update is being installed.

See Also

Reference

[GamerServicesDispatcher Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesDispatcher.InstallingTitleUpdate Event

Notifies the game when a Games for Windows - LIVE title update is being installed.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler<EventArgs> InstallingTitleUpdate
```

Remarks Gamer services and networking functionality is no longer available after this event has been raised: games should typically exit in response to **InstallingTitleUpdate**. If you are using the [GamerServicesComponent](#), this event is automatically hooked up to call [Exit](#), so you do not need to do anything special to handle title updates.

See Also

Reference

[GamerServicesDispatcher Class](#)

[GamerServicesDispatcher Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesNotAvailableException Class

Thrown if the [gamer services](#) system cannot be successfully initialized.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public class GamerServicesNotAvailableException : Exception
```

Remarks

The following are likely causes for this exception.

- If the current game uses the [XNA Framework Redistributable](#). This file does not include Games for Windows – LIVE functionality.
- If another Games for Windows – LIVE title is running already.

Important

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GamerServicesNotAvailableException](#).

See Also

Reference

[GamerServicesNotAvailableException Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GamerServicesNotAvailableException Members

The following tables list the members exposed by the `GamerServicesNotAvailableException` type.


Public Constructors

Name	Description
 GamerServicesNotAvailableException	Overloaded. Creates an instance of GamerServicesNotAvailableException .







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamerServicesNotAvailableException Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesNotAvailableException Constructor

Creates an instance of [GamerServicesNotAvailableException](#).

Overload List

Name	Description
GamerServicesNotAvailableException ()	Creates an empty instance of GamerServicesNotAvailableException .
GamerServicesNotAvailableException (SerializationInfo, StreamingContext)	Initializes a new instance of GamerServicesNotAvailableException with the specified streaming context.
GamerServicesNotAvailableException (String)	Initializes a new instance of GamerServicesNotAvailableException with the specified error message.
GamerServicesNotAvailableException (String, Exception)	Initializes a new instance of GamerServicesNotAvailableException with the specified error message and the inner exception.

See Also

Reference

[GamerServicesNotAvailableException Class](#)

[GamerServicesNotAvailableException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesNotAvailableException Constructor ()

Creates an empty instance of [GamerServicesNotAvailableException](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerServicesNotAvailableException ()
```

See Also

Reference

[GamerServicesNotAvailableException Class](#)

[GamerServicesNotAvailableException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesNotAvailableException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [GamerServicesNotAvailableException](#) with the specified streaming context.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected GamerServicesNotAvailableException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Describes the gamer services being accessed when the exception occurred.

context

Describes the stream where the exception occurred.

See Also

Reference

[GamerServicesNotAvailableException Class](#)

[GamerServicesNotAvailableException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GamerServicesNotAvailableException Constructor (String)

Initializes a new instance of [GamerServicesNotAvailableException](#) with the specified error message.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerServicesNotAvailableException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[GamerServicesNotAvailableException Class](#)

[GamerServicesNotAvailableException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesNotAvailableException Constructor (String, Exception)

Initializes a new instance of [GamerServicesNotAvailableException](#) with the specified error message and the inner exception.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerServicesNotAvailableException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The inner exception related to this exception.

See Also

Reference

[GamerServicesNotAvailableException Class](#)







[GamerServicesNotAvailableException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerServicesNotAvailableException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[GamerServicesNotAvailableException Class](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerServicesNotAvailableException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[GamerServicesNotAvailableException Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GamerZone Enumeration

This style of social gaming preferred by this Xbox LIVE member.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GamerZone
```

Members

Member name	Description
Family	Family-friendly game play.
Pro	Competitive game play.
Recreation	Non-competitive game play.
Underground	Alternative approach to game play.
Unknown	Unknown.

See Also

Reference

[GamerProfile.GamerZone Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide Class

Provides access to the Guide user interface. This interface contains message boxes, a text entry system, and other common interface elements.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class Guide
```

See Also

Reference

[Guide Members](#)






[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Guide Members

The following tables list the members exposed by the Guide type.

Public Properties



	Name	Description
	IsScreenSaverEnabled	Gets or sets the current state of the screen saver.
	IsTrialMode	Determines whether the game is running currently in limited trial mode.
	IsVisible	Determines whether a Guide user interface screen is active.
	NotificationPosition	Determines where notifications appear on the screen.
	SimulateTrialMode	Allows titles to simulate trial mode restrictions when testing using the Development configuration.

Public Methods

	Name	Description
	BeginShowKeyboardInput	Begins the process for displaying a dialog for keyboard input from a gamer.
	BeginShowMessageBox	Overloaded. Begins the process of displaying a message box with the specified parameters.
	BeginShowStorageDeviceSelector	Overloaded. Begins the process for displaying the storage device selector user interface.
	DelayNotifications	Delays system notifications for the specified amount of time.
	EndShowKeyboardInput	Ends the display of the keyboard input dialog box.
	EndShowMessageBox	Ends the display of a message box.
	EndShowStorageDeviceSelector	Ends the display of the storage selector user interface.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Show	Shows the guide user interface.
	ShowComposeMessage	Shows the Compose Message user interface.
	ShowFriendRequest	Shows the Friend Request user interface.
	ShowFriends	Shows the Friends user interface.
	ShowGameInvite	Displays the game invitation user interface.
	ShowGamerCard	Shows the Gamer Card user interface.
	ShowMarketplace	Displays the marketplace user interface.
	ShowMessages	Shows the Messages user interface.
	ShowParty	Shows the Xbox LIVE Party screen.
	ShowPartySessions	Shows the Play with Party Member screen.
	ShowPlayerReview	Shows the Player Review user interface.
	ShowPlayers	Shows the Players user interface.
	ShowSignIn	Shows the user interface a gamer uses for signing into Xbox LIVE.
	ToString	(Inherited from Object .)

Protected Methods

Name	Description
------	-------------

 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also













Reference

[Guide Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

Guide Methods

Public Methods

	Name	Description
	BeginShowKeyboardInput	Begins the process for displaying a dialog for keyboard input from a gamer.
	BeginShowMessageBox	Overloaded. Begins the process of displaying a message box with the specified parameters.
	BeginShowStorageDeviceSelector	Overloaded. Begins the process for displaying the storage device selector user interface.
	DelayNotifications	Delays system notifications for the specified amount of time.
	EndShowKeyboardInput	Ends the display of the keyboard input dialog box.
	EndShowMessageBox	Ends the display of a message box.
	EndShowStorageDeviceSelector	Ends the display of the storage selector user interface.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Show	Shows the guide user interface.
	ShowComposeMessage	Shows the Compose Message user interface.
	ShowFriendRequest	Shows the Friend Request user interface.
	ShowFriends	Shows the Friends user interface.
	ShowGameInvite	Displays the game invitation user interface.
	ShowGamerCard	Shows the Gamer Card user interface.
	ShowMarketplace	Displays the marketplace user interface.
	ShowMessages	Shows the Messages user interface.
	ShowParty	Shows the Xbox LIVE Party screen.
	ShowPartySessions	Shows the Play with Party Member screen.
	ShowPlayerReview	Shows the Player Review user interface.
	ShowPlayers	Shows the Players user interface.
	ShowSignIn	Shows the user interface a gamer uses for signing into Xbox LIVE.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Guide Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Guide.BeginShowKeyboardInput Method

Begins the process for displaying a dialog for keyboard input from a gamer. The display operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowKeyboardInput (
    PlayerIndex player,
    string title,
    string description,
    string defaultText,
    AsyncCallback callback,
    Object state
)
```

Parameters

player

Index of the player providing input. For Windows-based XNA games, the only valid option is **PlayerIndex.One**.

title

Title of the dialog box.

description

Text of the dialog box.

defaultText

Text displayed when the dialog box is initially shown.

callback

The method to be called once the asynchronous operation has finished.

state

A user-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
ArgumentException	<i>title</i> , <i>description</i> , or <i>defaultText</i> is larger than the maximum allowed string length of 260 characters.
ArgumentOutOfRangeException	<i>player</i> is not a valid value. For Windows-based XNA games, the only valid option is PlayerIndex.One .

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.BeginShowMessageBox Method

Begins the process of displaying a message box with the specified parameters. The display operation is performed asynchronously.

Overload List

Name	Description
Guide.BeginShowMessageBox (PlayerIndex, String, String, Generic IEnumerable, Int32, MessageBoxIcon, AsyncCallback, Object)	Begins the process of displaying a message box for a particular player.
Guide.BeginShowMessageBox (String, String, Generic IEnumerable, Int32, MessageBoxIcon, AsyncCallback, Object)	Begins the process of displaying a message box to which any user on the system can respond.

See Also

Reference

[GamerPrivileges.AllowCommunication Property](#)

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Guide.BeginShowMessageBox Method (PlayerIndex, String, String, Generic IEnumerable, Int32, MessageBoxIcon, AsyncCallback, Object)

Begins the process of displaying a message box for a particular player.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowMessageBox (
    PlayerIndex player,
    string title,
    string text,
    IEnumerable<string> buttons,
    int focusButton,
    MessageBoxIcon icon,
    AsyncCallback callback,
    Object state
)
```

Parameters

player

Target player of the message box. On Windows, the only valid option is **PlayerIndex.One**.

title

Title of the message box.

text

Text that will be displayed by the message box.

buttons

Captions for the message box buttons. The maximum number is three.

focusButton

Zero-based index specifying which button has the focus.

icon

Type of icon displayed in the message box.

callback

Method to be called once the asynchronous operation has finished.

state

User-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
InvalidOperationException	Either IsInitialized was not called previously, or a Guide user interface screen currently is active.

Remarks

This overload requires that the player for whom the message is targeted responds to the message.

★ Best Practice

You can check [GamerPrivileges.AllowCommunication](#) to determine if a player is allowed to send and receive communications before displaying a menu option that would call this method.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.BeginShowMessageBox Method (String, String, Generic IEnumerable, Int32, MessageBoxIcon, AsyncCallback, Object)

Begins the process of displaying a message box to which any user on the system can respond.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowMessageBox (
    string title,
    string text,
    IEnumerable<string> buttons,
    int focusButton,
    MessageBoxIcon icon,
    AsyncCallback callback,
    Object state
)
```

Parameters

title

Title of the message box.

text

Text that will be displayed by the message box.

buttons

Captions for the message box buttons. The maximum number is three.

focusButton

Zero-based index specifying which button has the focus.

icon

Type of icon displayed in the message box.

callback

Method to be called once the asynchronous operation has finished.

state

User-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
InvalidOperationException	Either IsInitialized was not called previously, or a Guide user interface screen currently is active.

Remarks

This overload allows any player on the system to respond to the message.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.BeginShowStorageDeviceSelector Method

Begins the process for displaying the storage device selector user interface.

Overload List

Name	Description
Guide.BeginShowStorageDeviceSelector (AsyncCallback, Object)	Begins the process for displaying the storage device selector user interface.
Guide.BeginShowStorageDeviceSelector (Int32, Int32, AsyncCallback, Object)	Begins the process for displaying the storage device selector user interface, specifying the size of the data to be written to the storage device.
Guide.BeginShowStorageDeviceSelector (PlayerIndex, AsyncCallback, Object)	Begins the process for displaying the storage device selector user interface, specifying the index of the player to be shown the user interface.
Guide.BeginShowStorageDeviceSelector (PlayerIndex, Int32, Int32, AsyncCallback, Object)	Begins the process for displaying the storage device selector user interface, specifying the index of the player to be shown the user interface, as well as the size of the data to be written to the storage device.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Guide.BeginShowStorageDeviceSelector Method (AsyncCallback, Object)

Begins the process for displaying the storage device selector user interface. The display operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowStorageDeviceSelector (
    AsyncCallback callback,
    Object state
)
```

Parameters

callback

The method to be called once the asynchronous operation has finished.

state

A user-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.BeginShowStorageDeviceSelector Method (Int32, Int32, AsyncCallback, Object)

Begins the process for displaying the storage device selector user interface, specifying the size of the data to be written to the storage device. The display operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowStorageDeviceSelector (  
    int sizeInBytes,  
    int directoryCount,  
    AsyncCallback callback,  
    Object state  
)
```

Parameters

sizeInBytes

Size, in bytes, of the data to write to the storage device.

directoryCount

Number of directories to write to the storage device.

callback

The method to be called once the asynchronous operation has finished.

state

A user-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.BeginShowStorageDeviceSelector Method (PlayerIndex, AsyncCallback, Object)

Begins the process for displaying the storage device selector user interface, specifying the index of the player to be shown the user interface. The display operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowStorageDeviceSelector (  
    PlayerIndex player,  
    AsyncCallback callback,  
    Object state  
)
```

Parameters

player

Index of the player being shown the user interface display. On Windows, the only valid option is **PlayerIndex.One**.

callback

The method to be called once the asynchronous operation has finished.

state

A user-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Remarks

The storage container, containing the data being saved, is associated with the gamertag of the specified *player*, not the current index of the *player*.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.BeginShowStorageDeviceSelector Method (PlayerIndex, Int32, Int32, AsyncCallback, Object)

Begins the process for displaying the storage device selector user interface, specifying the index of the player to be shown the user interface, as well as the size of the data to be written to the storage device. The display operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginShowStorageDeviceSelector (
    PlayerIndex player,
    int sizeInBytes,
    int directoryCount,
    AsyncCallback callback,
    Object state
)
```

Parameters

player

Index of the player being shown the user interface display. On Windows, the only valid option is **PlayerIndex.One**.

sizeInBytes

Size, in bytes, of the data to write to the storage device.

directoryCount

Number of directories to write to the storage device.

callback

The method to be called once the asynchronous operation has finished.

state

A user-created object uniquely identifying this request.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Remarks

The storage container, containing the data being saved, is associated with the gamertag of the specified *player*, not the current index of the *player*.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.DelayNotifications Method

Delays system notifications for the specified amount of time.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void DelayNotifications (  
    TimeSpan delay  
)
```

Parameters

delay

Time, in seconds, of the delay.

Calling this function while notifications are already delayed has no effect. The maximum delay is 120. If the specified duration exceeds the maximum, the maximum value is used.

Remarks

Use **DelayNotifications** to suppress pop-up notifications during critical times, such as cut scenes. Once the delay interval has passed, any deferred notifications are individually displayed in the order they were sent.

Titles cannot call **DelayNotifications** multiple times in a row. There must be an opportunity for the system application to display notifications to gamers.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.EndShowKeyboardInput Method

Ends the display of the keyboard input dialog box.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static string EndShowKeyboardInput (  
    IAsyncResult result  
)
```

Parameters

result

The [IAsyncResult](#) returned from [BeginShowKeyboardInput](#).

Return Value

Input received from the gamer.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.EndShowMessageBox Method

Ends the display of a message box.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Nullable<int> EndShowMessageBox (  
    IAsyncResult result  
)
```

Parameters

result

The [IAsyncResult](#) returned from [BeginShowMessageBox](#).

Return Value

Remarks

This call blocks until the display operation is finished. Typically, you call this from the callback function specified in the *callback* argument of [BeginShowMessageBox](#).

Each call to [EndShowMessageBox](#) must match an earlier call to [BeginShowMessageBox](#).

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.EndShowStorageDeviceSelector Method

Ends the display of the storage selector user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static StorageDevice EndShowStorageDeviceSelector (
    IAsyncResult result
)
```

Parameters

result

The [IAsyncResult](#) returned from [BeginShowStorageDeviceSelector](#).

Return Value

Storage device selected by the gamer, or **null** if the user cancelled the selection.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.Show Method

Note

This method is available only when developing for Zune.

Shows the guide user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Show ()
```

Exceptions

Exception type	Condition
GuideAlreadyVisibleException	The Guide is already being displayed on the target device.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

[PlatformsZune](#)

Guide.ShowComposeMessage Method

Shows the **Compose Message** user interface. A gamer uses this user interface to write messages to other gamers.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowComposeMessage (
    PlayerIndex player,
    string text,
    IEnumerable<Gamer> recipients
)
```

Parameters

player

Gamer composing the message.

text

Text of the message.

recipients

Collection of gamers receiving the message.

If this is **null**, the **Select Gamertag** user interface is displayed.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	For Windows applications, this exception is thrown if <i>player</i> is not PlayerIndex.One .
ArgumentException	The message text is null , empty, or exceeds 200 characters. This exception is also thrown if any recipients have a value of null or if there are more than 100 recipients.

Remarks

★ Best Practice

You can check [GamerPrivileges.AllowCommunication](#) to determine if a player is allowed to send and receive communication s prior to displaying a menu option that would call this method.

See Also

Reference

[GamerPrivileges.AllowCommunication Property](#)

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowFriendRequest Method

Shows the **Friend Request** user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowFriendRequest (  
    PlayerIndex player,  
    Gamer gamer  
)
```

Parameters

player

Index of the player making the request.

The profile specified by *player* must be signed in when calling this function.

gamer

Recipient of the request.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowFriends Method

Shows the **Friends** user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowFriends (  
    PlayerIndex player  
)
```

Parameters

player

Index of the player making the request.

The profile specified by *player* must be signed in when calling this function.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowGameInvite Method

Displays the game invitation user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowGameInvite (  
    PlayerIndex player,  
    IEnumerable<Gamer> recipients  
)
```

Parameters

player

Index of the player initiating the invitation.

recipients

List of gamers who receive the invitation. If **null**, the Guide will prompt the player to specify a recipient.

Remarks

See [How To: Add Support for Game Invitations](#) for more information about adding game invitation support to your title.

★ Best Practice

You can check [GamerPrivileges.AllowCommunication](#) to determine if a player is allowed to send and receive communication s prior to displaying a menu option that would call this method.

See Also

Reference

[NetworkSession.InviteAccepted Event](#)

[NetworkSession.JoinInvited Method](#)

[NetworkSession.BeginJoinInvited Method](#)

[NetworkSession.EndJoinInvited Method](#)

[GamerPrivileges.AllowCommunication Property](#)

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowGamerCard Method

Shows the **Gamer Card** user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowGamerCard (  
    PlayerIndex player,  
    Gamer gamer  
)
```

Parameters

player

Index of the player making the request.

The profile specified by *player* must be signed in when calling this function.

gamer

Owner of the gamer card being requested for display.

Remarks

★ Best Practice

You can check [GamerPrivileges.AllowProfileViewing](#) to determine if a player is allowed to send and receive communications prior to displaying a menu option that would call this method.

See Also

Reference

[GamerPrivileges.AllowProfileViewing](#) Property

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowMarketplace Method

Displays the marketplace user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowMarketplace (  
    PlayerIndex player  
)
```

Parameters

player

Index of the controller for the player making the request.

Remarks

★ Best Practice

- You can check [GamerPrivileges.AllowPurchaseContent](#) to determine if a player is allowed to purchase content before displaying a menu option that would call this method.
- When a trial game is exiting, you can present a marketplace offer using the [Guide.ShowMarketplace](#) method. You can check if a game is in trial mode using [Guide.IsTrialMode](#), and write code that is performed on game exit in the [Game.Exit](#) method.

See Also

Tasks

[How To: Detect or Simulate Trial Mode and Present a Marketplace Offer](#)

Reference

[GamerPrivileges.AllowPurchaseContent](#) Property

[Guide.SimulateTrialMode](#) Property

[Guide.IsTrialMode](#) Property

[Guide](#) Class

[Guide](#) Members

[Microsoft.Xna.Framework.GamerServices](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowMessages Method

Shows the **Messages** user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowMessages (  
    PlayerIndex player  
)
```

Parameters

player

Index of the player making the request.

The profile specified by *player* must be signed in when calling this function.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowParty Method

Shows the Xbox LIVE Party screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowParty (  
    PlayerIndex player  
)
```

Parameters

player

The player index for whom to bring up the Xbox LIVE Party screen.

Remarks

Shows the Xbox LIVE Party screen, which allows players to invite others to the party, invite party members to the current game session, or to chat with party members.

Note

For LIVE-enabled platforms without LIVE Party support, the **Friends** screen will be shown instead.

See Also

Tasks

[How To: Add LIVE Party Support](#)

[How To: Add Support for Game Invitations](#)

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowPartySessions Method

Shows the **Play with Party Member** screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowPartySessions (  
    PlayerIndex player  
)
```

Parameters

player

The player index for whom the list of game sessions with party members will be shown.

Remarks

This method brings up the **Play with Party Member** screen, which allows players to join game sessions with fellow party members in the current title.

If the player is not currently in a party, this method will show the Friends screen instead.

See Also

Tasks

[How To: Add LIVE Party Support](#)

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowPlayerReview Method

Shows the **Player Review** user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowPlayerReview (  
    PlayerIndex player,  
    Gamer gamer  
)
```

Parameters

player

Index of the player making the request.

The profile specified by *player* must be signed in when calling this function.

gamer

Player being reviewed.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowPlayers Method

Shows the **Players** user interface.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowPlayers (  
    PlayerIndex player  
)
```

Parameters

player

Index of the player making the request.

The profile specified by *player* must be signed in when calling this function.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.ShowSignIn Method

Shows the user interface a gamer uses for signing into Xbox LIVE.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void ShowSignIn (
    int paneCount,
    bool onlineOnly
)
```

Parameters

paneCount

Number of panes displayed at sign in. Each pane accepts a single gamer.

On Windows, only a single pane can be displayed. On Xbox 360, valid values are 1, 2, and 4.

onlineOnly

Determines the profile types being displayed. **true** if only online profiles are displayed; **false** otherwise. If *onlineOnly* is **true**, local gamers can sign in as guests of a profile currently signed in.

Exceptions

Exception type	Condition
ArgumentException	The <i>paneCount</i> argument is invalid. The valid options for this parameter are 1, 2, and 4.

See Also

Reference

[Guide Class](#)











[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide Properties

Public Properties

	Name	Description
 	IsScreenSaverEnabled	Gets or sets the current state of the screen saver.
 	IsTrialMode	Determines whether the game is running currently in limited trial mode.
 	IsVisible	Determines whether a Guide user interface screen is active.
 	NotificationPosition	Determines where notifications appear on the screen.
 	SimulateTrialMode	Allows titles to simulate trial mode restrictions when testing using the Development configuration.

See Also

Reference

[Guide Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Guide.IsScreenSaverEnabled Property

Gets or sets the current state of the screen saver. This property has no effect for Windows games.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsScreenSaverEnabled { get; set; }
```

Property Value

true if the screen saver is enabled; **false** otherwise.

Remarks

The **IsScreenSaverEnabled** property is a per-game property. Disabling the screen saver for this game does not permanently disable the screen saver. Use this method when displaying cutscenes or during other periods where the gamer could become inactive for a long period of time.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.IsTrialMode Property

Determines whether the game is running currently in limited trial mode.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsTrialMode { get; set; }
```

Property Value

true if the game is running currently in limited trial mode; **false** otherwise.

Remarks

★Best Practice

When a trial game is exiting, you can present a marketplace offer using the [Guide.ShowMarketplace](#) method. You can write code that is performed on game exit in the [Game.Exit](#) method.

Limited Trial Mode

While in trial mode, games will be unable to use the Xbox LIVE servers to connect to other machines over the Internet for player match or ranked network sessions. Note that this does not affect system link sessions, which are always available.

The [Guide.IsTrialMode](#) property always returns **true** when a game begins. [Guide.IsTrialMode](#) may change to **false** shortly after game launch if a license is detected when a player signs in, or it could change at any time after game launch if a license is acquired.

XNA Creators Club games can be launched in simulated trial mode by setting the [Guide.SimulateTrialMode](#) property to **true** when the game starts. During development, when XNA Creators Club games are running on an Xbox 360 console, there is also an option in the dash to launch a game in simulated trial mode.

Indie games downloaded from the Xbox LIVE Marketplace will run in limited trial mode until one of the signed-in gamers acquires a license for the game. Detection of a game license could occur as part of game startup, in response to an in-game purchase, or as part of user signing in.

For Xbox LIVE Indie Games running in trial mode, gameplay will be limited to a short trial period. At the end of the gameplay period, the player will see an expiration screen. The player can then either exit the game via the **B** or **Back** button or use the **X** button to unlock the game. If the player chooses to unlock the game, a **Marketplace** screen will be displayed. The player can then buy the full version of the game.

While the expiration screen shows, [Game.IsActive](#) will return **false**, and all input to the game will be blocked. Rendering can continue. However, the expiration screen obscures most of the rendering process, and any controller vibration will be stopped.

★Best Practice

It is generally a good idea to pause the game when [Game.IsActive](#) is **false**.

For XNA Creators Club games running in simulated trial mode there will be no time limitation applied to gameplay.

See Also

Tasks

[How To: Detect or Simulate Trial Mode and Present a Marketplace Offer](#)

[How To: Pause a Game](#)

Reference

[Guide.ShowMarketplace Method](#)

[Guide.SimulateTrialMode Property](#)

[SignedInGamer.GameDefaults Property](#)

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.IsVisible Property

Determines whether a Guide user interface screen is active.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsVisible { get; set; }
```

Property Value

true if a user interface screen is displayed; **false** otherwise. It is not possible to bring up another Guide user interface screen, if **IsVisible** is **true**.

Exceptions

Exception type	Condition
InvalidOperationException	The gamer services functionality must be initialized before this property is accessed.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Guide.NotificationPosition Property

Determines where notifications appear on the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NotificationPosition NotificationPosition { get; set; }
```

Property Value

Defines the positioning of the notification message box.

See Also

Reference

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Guide.SimulateTrialMode Property

Allows titles to simulate trial mode restrictions when testing using the **Development** configuration. Setting this property to **true** in the game constructor will artificially force [IsTrialMode](#) to return **true**.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool SimulateTrialMode { get; set; }
```

Property Value

true to simulate trial mode restrictions in a **Development** configuration; **false** otherwise.

Remarks

You can set **SimulateTrialMode** to **false** at some later point (perhaps calling the ShowMarketplace API at the same time) to simulate purchasing the full game.

Limited Trial Mode

While in trial mode, games will be unable to use the Xbox LIVE servers to connect to other machines over the Internet for player match or ranked network sessions. Note that this does not affect system link sessions, which are always available.

The [Guide.IsTrialMode](#) property always returns **true** when a game begins. [Guide.IsTrialMode](#) may change to **false** shortly after game launch if a license is detected when a player signs in, or it could change at any time after game launch if a license is acquired.

XNA Creators Club games can be launched in simulated trial mode by setting the [Guide.SimulateTrialMode](#) property to **true** when the game starts. During development, when XNA Creators Club games are running on an Xbox 360 console, there is also an option in the dash to launch a game in simulated trial mode.

Indie games downloaded from the Xbox LIVE Marketplace will run in limited trial mode until one of the signed-in gamers acquires a license for the game. Detection of a game license could occur as part of game startup, in response to an in-game purchase, or as part of user signing in.

For Xbox LIVE Indie Games running in trial mode, gameplay will be limited to a short trial period. At the end of the gameplay period, the player will see an expiration screen. The player can then either exit the game via the **B** or **Back** button or use the **X** button to unlock the game. If the player chooses to unlock the game, a **Marketplace** screen will be displayed. The player can then buy the full version of the game.

While the expiration screen shows, [Game.IsActive](#) will return **false**, and all input to the game will be blocked. Rendering can continue. However, the expiration screen obscures most of the rendering process, and any controller vibration will be stopped.

★ Best Practice

It is generally a good idea to pause the game when [Game.IsActive](#) is **false**.

For XNA Creators Club games running in simulated trial mode there will be no time limitation applied to gameplay.

See Also

Tasks

[How To: Detect or Simulate Trial Mode and Present a Marketplace Offer](#)

Reference

[Guide.ShowMarketplace Method](#)

[Guide.IsTrialMode Property](#)

[NetworkGamer.IsGuest Property](#)

[Guide Class](#)

[Guide Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GuideAlreadyVisibleException Class

Thrown if an attempt is made to display a component of the Guide user interface when a Guide component is already displayed.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public class GuideAlreadyVisibleException : Exception
```

See Also

Reference

[GuideAlreadyVisibleException Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GuideAlreadyVisibleException Members

The following tables list the members exposed by the GuideAlreadyVisibleException type.


Public Constructors

Name	Description
 GuideAlreadyVisibleException	Overloaded. Creates an instance of GuideAlreadyVisibleException.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GuideAlreadyVisibleException Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GuideAlreadyVisibleException Constructor

Creates an instance of **GuideAlreadyVisibleException**.

Overload List

Name	Description
GuideAlreadyVisibleException ()	Initializes an empty instance of GuideAlreadyVisibleException .
GuideAlreadyVisibleException (SerializationInfo, StreamingContext)	Initializes an instance of GuideAlreadyVisibleException with the specified streaming context.
GuideAlreadyVisibleException (String)	Initializes an instance of GuideAlreadyVisibleException with the specified error message.
GuideAlreadyVisibleException (String, Exception)	Initializes an instance of GuideAlreadyVisibleException with the specified error message and related inner exception.

See Also

Reference

[GuideAlreadyVisibleException Class](#)

[GuideAlreadyVisibleException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

GuideAlreadyVisibleException Constructor ()

Initializes an empty instance of **GuideAlreadyVisibleException**.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GuideAlreadyVisibleException ()
```

See Also

Reference

[GuideAlreadyVisibleException Class](#)

[GuideAlreadyVisibleException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GuideAlreadyVisibleException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Initializes an instance of **GuideAlreadyVisibleException** with the specified streaming context.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected GuideAlreadyVisibleException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Describes the displayed Guide component when the exception occurred.

context

Describes the stream where the exception occurred.

See Also

Reference

[GuideAlreadyVisibleException Class](#)

[GuideAlreadyVisibleException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GuideAlreadyVisibleException Constructor (String)

Initializes an instance of **GuideAlreadyVisibleException** with the specified error message.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GuideAlreadyVisibleException (  
    string message  
)
```

Parameters

message

A message describing the error.

See Also

Reference

[GuideAlreadyVisibleException Class](#)

[GuideAlreadyVisibleException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GuideAlreadyVisibleException Constructor (String, Exception)

Initializes an instance of **GuideAlreadyVisibleException** with the specified error message and related inner exception.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GuideAlreadyVisibleException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message describing the error.

innerException

The inner exception related to this exception.

See Also

Reference

[GuideAlreadyVisibleException Class](#)







[GuideAlreadyVisibleException Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GuideAlreadyVisibleException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[GuideAlreadyVisibleException Class](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

GuideAlreadyVisibleException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[GuideAlreadyVisibleException Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

MessageBoxIcon Enumeration

Defines the different icons for a message box.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum MessageBoxIcon
```

Members

Member name	Description
Alert	Displays the Alert icon.
Error	Displays the Error icon.
None	No icon is displayed.
Warning	Displays the Warning icon.

See Also

Reference

[Guide.BeginShowMessageBox Method](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NotificationPosition Enumeration

Determines where notifications appear on the screen. Used to specify the notification position for the [NotificationPosition](#) property.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum NotificationPosition
```

Members

Member name	Description
BottomCenter	Positions the message box at the bottom of the screen and centered.
BottomLeft	Positions the message box at the bottom-left of the screen.
BottomRight	Positions the message box at the bottom-right of the screen.
Center	Positions the message box at the center of the screen.
CenterLeft	Positions the message box at the center of the screen and left-aligned.
CenterRight	Positions the message box at the center of the screen and right-aligned.
TopCenter	Positions the message box at the top of the screen and centered.
TopLeft	Positions the message box at the top-left of the screen.
TopRight	Positions the message box at the top-right of the screen.

See Also

Reference

[Guide.NotificationPosition Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RacingCameraAngle Enumeration

Indicates which camera angle this gamer prefers to use in racing games.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum RacingCameraAngle
```

Members

Member name	Description
Back	Traditional third-person camera view from behind the car.
Front	Camera view from in front of the car. The car itself is not visible.
Inside	Camera view from inside the car, looking through the windscreen.

See Also

Reference

[GameDefaults.RacingCameraAngle Property](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInEventArgs Class

Represents the arguments passed to a [SignedIn](#) event. These arguments are passed to event handlers when a new gamer signs in. This class contains the [SignedInGamer](#) instance that recently signed in.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class SignedInEventArgs : EventArgs
```

See Also

Reference

[SignedInEventArgs Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


SignedInEventArgs Members

The following tables list the members exposed by the SignedInEventArgs type.






Public Constructors

	Name	Description
	SignedInEventArgs	Creates a new instance of SignedInEventArgs .



Public Properties

	Name	Description
	Gamer	Gets the gamer that just signed in.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SignedInEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInEventArgs Fields

See Also

Reference

[SignedInEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInEventArgs Constructor

Creates a new instance of [SignedInEventArgs](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedInEventArgs (  
    SignedInGamer gamer  
)
```

Parameters

gamer

The gamer that just signed in.

Exceptions

Exception type	Condition
ArgumentNullException	<i>gamer</i> is null .

See Also

Reference

[SignedInEventArgs Class](#)






[SignedInEventArgs Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[SignedInEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInEventArgs Properties

Public Properties

	Name	Description
	Gamer	Gets the gamer that just signed in.

See Also

Reference

[SignedInEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInEventArgs.Gamer Property

Gets the gamer that just signed in.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedInGamer Gamer { get; }
```

Property Value

The gamer that just signed in.

See Also

Reference

[SignedInEventArgs Class](#)

[SignedInEventArgs Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer Class

Represents a gamer (a profile that has an associated gamertag) on the local system.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SignedInGamer : Gamer
```

Remarks

Gamers can be either on a local system or in a multiplayer session. For more information about gamers in a multiplayer session, see [NetworkGamer](#).

See Also

Reference

[SignedInGamer Members](#)













[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










SignedInGamer Members

The following tables list the members exposed by the SignedInGamer type.



Public Properties

Name	Description
 Avatar	Description data for the avatar that represents the gamer.
 GameDefaults	Describes a gamer's preferred settings.
 Gamertag	(Inherited from Gamer .)
 IsDisposed	(Inherited from Gamer .)
 IsGuest	Determines whether the gamer is the guest of an Xbox LIVE-enabled profile.
 IsSignedInToLive	Determines whether the gamer has an Xbox LIVE-enabled profile.
 PartySize	Gets the current party size.
 PlayerIndex	Gets the index of the gamer.
 Presence	Gets an object that may be used to set the rich presence state for this gamer.
 Privileges	Describes what operations a gamer is allowed to perform.
 SignedInGamers	(Inherited from Gamer .)
 Tag	(Inherited from Gamer .)




Public Methods

Name	Description
 BeginGetProfile	(Inherited from Gamer .)
 EndGetProfile	(Inherited from Gamer .)
 Equals	(Inherited from Object .)
 GetFriends	Reads the friends list of this local gamer. This includes both the gamertags of the friends and their current presence information.
 GetHashCode	(Inherited from Object .)
 GetProfile	(Inherited from Gamer .)
 GetType	(Inherited from Object .)
 IsFriend	Queries whether the specified gamer is a friend of this local gamer.
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Public Events

Name	Description
 AvatarChanged	Occurs when a gamer's avatar changes.
 SignedIn	Occurs when a new gamer signs into the local system.
 SignedOut	Occurs when a gamer signs out on the local system.

See Also










Reference

[SignedInGamer Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamer Methods

Public Methods

	Name	Description
	BeginGetProfile	(Inherited from Gamer .)
	EndGetProfile	(Inherited from Gamer .)
	Equals	(Inherited from Object .)
	GetFriends	Reads the friends list of this local gamer. This includes both the gamertags of the friends and their current presence information.
	GetHashCode	(Inherited from Object .)
	GetProfile	(Inherited from Gamer .)
	GetType	(Inherited from Object .)
	IsFriend	Queries whether the specified gamer is a friend of this local gamer.
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SignedInGamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamer.GetFriends Method

Reads the friends list of this local gamer. This includes both the gamertags of the friends and their current presence information.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FriendCollection GetFriends ()
```

Return Value

The friends of this local gamer.

Exceptions

Exception type	Condition
ObjectDisposedException	This object has already been disposed.
GamerPrivilegeException	A signed-in gamer profile is required to perform this operation. A LIVE profile may also be required. There are no profiles currently signed in, or the profile is not signed in to LIVE.

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.IsFriend Method

Queries whether the specified gamer is a friend of this local gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsFriend (
    Gamer gamer
)
```

Parameters

gamer

The gamer to query.

Return Value

true if the specified gamer is a friend of the local gamer; **false** otherwise.

Exceptions

Exception type	Condition
ArgumentNullException	<i>gamer</i> is null .
GamerPrivilegeException	A signed in gamer profile is required to perform this operation. A LIVE profile may also be required. There are no profiles currently signed in, or the profile is not signed in to LIVE.
ObjectDisposedException	The <i>gamer</i> object has already been disposed.

See Also

Reference

[SignedInGamer Class](#)













[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer Properties

Public Properties

	Name	Description
	Avatar	Description data for the avatar that represents the gamer.
	GameDefaults	Describes a gamer's preferred settings.
	Gamertag	(Inherited from Gamer .)
	IsDisposed	(Inherited from Gamer .)
	IsGuest	Determines whether the gamer is the guest of an Xbox LIVE-enabled profile.
	IsSignedInToLive	Determines whether the gamer has an Xbox LIVE-enabled profile.
	PartySize	Gets the current party size.
	PlayerIndex	Gets the index of the gamer.
	Presence	Gets an object that may be used to set the rich presence state for this gamer.
	Privileges	Describes what operations a gamer is allowed to perform.
	SignedInGamers	(Inherited from Gamer .)
	Tag	(Inherited from Gamer .)

See Also

Reference

[SignedInGamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamer.Avatar Property

Description data for the avatar that represents the gamer.

Windows Specific Information

Avatar-related APIs are exposed on Windows to ease cross-platform development between Windows and the Xbox 360. All avatar-related methods return default values, and rendering methods do not draw anything to the screen.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AvatarDescription Avatar { get; }
```

Property Value

Description data of the related avatar.

Remarks

It is possible that a newly-created random [AvatarDescription](#) object could be an invalid avatar description buffer. Possible scenarios where this could occur include:

- The associated avatar data, stored on the local storage of the signed-in gamer, is corrupted.
- The signed-in gamer currently has no avatar.

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SignedInGamer.GameDefaults Property

Describes a gamer's preferred settings.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GameDefaults GameDefaults { get; }
```

Property Value

The gamer's preferred settings.

RemarksWherever possible, games should default to these values.

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.IsGuest Property

Determines whether the gamer is the guest of an Xbox LIVE-enabled profile.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsGuest { get; }
```

Property Value

true if the gamer is a guest; **false** otherwise.

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.IsSignedInToLive Property

Determines whether the gamer has an Xbox LIVE-enabled profile.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsSignedInToLive { get; }
```

Property Value

true if the gamer is able to sign in to Xbox LIVE; **false** otherwise.

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.PartySize Property

Gets the current party size.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PartySize { get; set; }
```

Property Value

The current number of members in this signed-in gamer's LIVE Party.

Remarks

This is a read-only property. You cannot use it to set the party size. On platforms that do not support LIVE parties, the value of this property is always zero.

See Also

Tasks

[How To: Add LIVE Party Support](#)

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.PlayerIndex Property

Gets the index of the gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlayerIndex PlayerIndex { get; }
```

Property Value

Index of the gamer.

Exceptions

Exception type	Condition
ObjectDisposedException	Thrown if the associated profile is no longer valid. For example, the profile may have signed out.

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.Presence Property

Gets an object that may be used to set the rich presence state for this gamer.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPresence Presence { get; }
```

Property Value

An object that may be used to set the rich presence state for this gamer.

Remarks

Zune Specific Information

Setting presence information is not supported on the Zune platform. On the Zune platform, setting presence information for a signed-in gamer will not change the presence information on Xbox LIVE.

Note

A game title will not be displayed in a player's presence information on the Xbox LIVE service until that game passes peer review. Once a game passes peer review, the presence information for a player shows three things: a confirmation the player is playing an Xbox LIVE Indie Game, the game's title, and the presence information set by [SignedInGamer.Presence](#). For example:

```
Xbox LIVE Indie Game  
My Game Title  
Fighting the Boss
```

While a game is in development, but before the game passes peer review, the presence information indicates the developer is using the XNA Creators Club, as shown:

```
XNA Creators Club  
creators.xna.com  
Fighting the Boss
```

See Also

Tasks

[How To: Add Presence Information](#)

Reference

[Gamer.SignedInGamers Property](#)

[GamerPresence Class](#)

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.Privileges Property

Describes what operations a gamer is allowed to perform.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerPrivileges Privileges { get; }
```

Property Value

The operations a gamer is allowed to perform.

Remarks **Privileges** can be defined by parental control settings, and also will be set automatically in response to things like age, region, and whether the gamer has a LIVE Gold or LIVE Silver membership. Games do not need to explicitly check privileges, because a [GamerPrivilegeException](#) will be thrown if an unsupported operation is selected. However, these privilege bits may be useful to detect that an operation is unavailable before calling it so as to grey-out the relevant menu option.

See Also

Reference

[SignedInGamer Class](#)






[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer Events

Public Events

	Name	Description
	AvatarChanged	Occurs when a gamer's avatar changes.
 	SignedIn	Occurs when a new gamer signs into the local system.
 	SignedOut	Occurs when a gamer signs out on the local system.

See Also

Reference

[SignedInGamer Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamer.AvatarChanged Event

Occurs when a gamer's avatar changes.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<EventArgs> AvatarChanged
```

See Also

Reference

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SignedInGamer.SignedIn Event

Occurs when a new gamer signs into the local system.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler<SignedInEventArgs> SignedIn
```

Remarks

This event is raised during calls to [Update](#). When this event is raised, the [LocalGamers](#) collection will contain the newly signed-in gamer.

Behavior of Local Sessions

XNA Game Studio 3.0 introduces a change in the behavior of network sessions when local profiles sign out in the middle of a session. In previous versions of XNA Game Studio, a session ended whenever a player signed out. Now the session continues even if one or more profiles sign out. Signed-out players are removed from the session, but the session ends only if no suitable players remain in the session.

See Also

Reference

[SignedOut](#)

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamer.SignedOut Event

Occurs when a gamer signs out on the local system.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler<SignedOutEventArgs> SignedOut
```

Remarks

This event is raised during calls to [Update](#). When this event is raised, the [LocalGamers](#) collection will no longer contain the signed-out gamer.

Behavior of Local Sessions

XNA Game Studio 3.0 introduces a change in the behavior of network sessions when local profiles sign out in the middle of a session. In previous versions of XNA Game Studio, a session ended whenever a player signed out. Now the session continues even if one or more profiles sign out. Signed-out players are removed from the session, but the session ends only if no suitable players remain in the session.

See Also

Reference

[SignedIn](#)

[SignedInGamer Class](#)

[SignedInGamer Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedInGamerCollection Class

Represents a collection of gamers on the local system.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SignedInGamerCollection : GamerCollection<SignedInGamer>
```

See Also

Reference

[SignedInGamerCollection Members](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


SignedInGamerCollection Members

The following tables list the members exposed by the SignedInGamerCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	Overloaded. Gets a specific SignedInGamer object.



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[SignedInGamerCollection Class](#)



[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamerCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also



Reference

[SignedInGamerCollection Class](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamerCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection .)
	Item	Overloaded. Gets a specific SignedInGamer object.

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection .)

See Also

Reference

[SignedInGamerCollection Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamerCollection.Item Property

Gets a specific [SignedInGamer](#) object.

Overload List

Name	Description
SignedInGamerCollection.Item (PlayerIndex)	Gets a specific local gamer based on that player's PlayerIndex on the local system.
SignedInGamerCollection.Item (Int32)	(Inherited from ReadOnlyCollection .)

See Also

Reference

[SignedInGamerCollection Class](#)

[SignedInGamerCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedInGamerCollection.Item Property (PlayerIndex)

Gets a specific local gamer based on that player's [PlayerIndex](#) on the local system.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedInGamer this [  
    PlayerIndex index  
] { get; }
```

Property Value

The local gamer at the specified [PlayerIndex](#).

See Also

Reference

[SignedInGamerCollection Class](#)

[SignedInGamerCollection Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedOutEventArgs Class

Represents the arguments passed to a [SignedOut](#) event. These arguments are passed to event handlers when a gamer signs out. This class contains the [SignedInGamer](#) instance that recently signed out.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class SignedOutEventArgs : EventArgs
```

See Also

Reference

[SignedOutEventArgs Members](#)


[Microsoft.Xna.Framework.GamerServices Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


SignedOutEventArgs Members

The following tables list the members exposed by the SignedOutEventArgs type.






Public Constructors

	Name	Description
	SignedOutEventArgs	Creates a new instance of SignedOutEventArgs .



Public Properties

	Name	Description
	Gamer	Gets the gamer that just signed out.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SignedOutEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedOutEventArgs Fields

See Also

Reference

[SignedOutEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedOutEventArgs Constructor

Creates a new instance of [SignedOutEventArgs](#).

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedOutEventArgs (  
    SignedInGamer gamer  
)
```

Parameters

gamer

The gamer that just signed out.

Exceptions

Exception type	Condition
ArgumentNullException	<i>gamer</i> is null .

See Also

Reference

[SignedOutEventArgs Class](#)






[SignedOutEventArgs Members](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SignedOutEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[SignedOutEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedOutEventArgs Properties

Public Properties

	Name	Description
	Gamer	Gets the gamer that just signed out.

See Also

Reference

[SignedOutEventArgs Class](#)

[Microsoft.Xna.Framework.GamerServices Namespace](#)

SignedOutEventArgs.Gamer Property

Gets the gamer that just signed out.

Namespace: Microsoft.Xna.Framework.GamerServices

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedInGamer Gamer { get; }
```

Property Value

The gamer that just signed out.

See Also

Reference

[SignedOutEventArgs Class](#)

[SignedOutEventArgs Members](#)













[Microsoft.Xna.Framework.GamerServices Namespace](#)





























Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics Namespace

Contains low-level application programming interface (API) methods that take advantage of hardware acceleration capabilities to display 3D objects.

Classes

	Name	Description
	BasicDirectionalLight	Provides methods for manipulating the directional lights used by the BasicEffect class.
	BasicEffect	Represents shader model 1.1 Effect that has optional support for vertex colors, texturing, and lighting.
	ClipPlane	Represents the device clipping plane.
	ClipPlaneCollection	Manipulates a collection of ClipPlane objects.
	CompilationFailedException	The exception that is thrown if the compilation of an effect fails.
	CompilerIncludeHandler	The abstract base class for custom compiler include file handlers.
	DepthStencilBuffer	Queries and prepares depth stencil buffers.
	DeviceLostException	The exception that is thrown when the device has been lost, but cannot be reset at this time. Therefore, rendering is not possible.
	DeviceNotResetException	The exception that is thrown when the device has been lost, but can be reset at this time.
	DeviceNotSupportedException	The exception that is thrown when the graphics device does not support the requested capabilities.
	DeviceStillDrawingException	The error that is thrown when the device is still drawing.
	DriverInternalErrorException	The exception that is thrown when an internal driver error occurs. Applications should destroy and recreate the device when receiving this error.
	DynamicIndexBuffer	Describes the rendering order of the vertices in a vertex buffer.
	DynamicVertexBuffer	Represents a list of 3D vertices to be streamed to the graphics device.
	Effect	Used to set and query effects and choose techniques.
	EffectAnnotation	Represents an annotation to an EffectParameter .
	EffectAnnotationCollection	Manipulates a collection of EffectAnnotation objects.
	EffectFunction	Represents a function on an Effect .
	EffectFunctionCollection	Manipulates a collection of EffectFunction objects.
	EffectParameter	Represents an Effect parameter.
	EffectParameterBlock	Represents an EffectParameter state block.
	EffectParameterCollection	Manipulates a collection of EffectParameter objects.
	EffectPass	Represents an effect pass.

 EffectPassCollection	Manipulates a collection of EffectPass objects.
 EffectPool	Allows applications to share resources between effects.
 EffectTechnique	Represents an effect technique.
 EffectTechniqueCollection	Manipulates a collection of EffectTechnique objects.
 GammaRamp	Contains red, green, and blue ramp data.
 GraphicsAdapter	Provides methods to retrieve and manipulate graphics adapters.
 GraphicsDevice	Performs primitive-based rendering, creates resources, handles system-level variables, adjusts gamma ramp levels, and creates shaders.
 GraphicsDeviceCapabilities	Represents the capabilities of the hardware.
 GraphicsResource	Queries and prepares resources.
 IndexBuffer	Describes the rendering order of the vertices in a vertex buffer.
 Model	Represents a 3D model composed of multiple ModelMesh objects which may be moved independently.
 ModelBone	Represents bone data for a model.
 ModelBoneCollection	Represents a set of bones associated with a model.
 ModelErrorCollection	Represents a collection of effects associated with a model.
 ModelMesh	Represents a mesh that is part of a Model .
 ModelMeshCollection	Represents a collection of ModelMesh objects.
 ModelMeshPart	Represents a batch of geometry information to submit to the graphics device during rendering. Each ModelMeshPart is a subdivision of a ModelMesh object. The ModelMesh class is split into multiple ModelMeshPart objects, typically based on material information.
 ModelMeshPartCollection	Represents a collection of ModelMeshPart objects.
 OcclusionQuery	Used to perform an occlusion query against the latest drawn objects.
 OutOfVideoMemoryException	The exception that is thrown when Direct3D does not have enough display memory to perform the operation.
 PixelShader	Encapsulates the functionality of a pixel shader.
 PresentationParameters	Contains presentation parameters.
 RenderState	Defines the render state of a graphics device.
 RenderTarget	Represents a resource that will be written to at the end of a render pass. This is the base class for RenderTarget2D and RenderTargetCube .
 RenderTarget2D	Represents a 2D texture resource that will be written to at the end of a render pass.
 RenderTargetCube	Represents a cubic texture resource that will be written to at the end of a render pass.
 ResolveTexture2D	Represents a 2D grid of texels.
 ResourceCreatedEventArgs	Contains event data.





























ResourceDestroyedEventArgs	Arguments for a ResourceDestroyed event.
SamplerState	Contains sampler states for the device.
SamplerStateCollection	Collection of SamplerState objects.
ShaderCompiler	Compiles and decompiles high-level shader language (HLSL) shaders.
ShaderConstant	Describes a shader constant.
ShaderConstantCollection	Manipulates a collection of ShaderConstant objects.
ShaderConstantTable	Contains the variables that are used by high-level language shaders and effects.
SpriteBatch	Enables a group of sprites to be drawn using the same settings.
SpriteFont	Represents a font texture.
StateBlock	Encapsulates render states.
Texture	Represents a texture resource.
Texture2D	Represents a 2D grid of texels.
Texture3D	Represents a 3D volume of texels.
TextureCollection	Represents a collection of Texture objects.
TextureCube	Represents a set of six 2D textures, one for each face of a cube.
VertexBuffer	Represents a list of 3D vertices to be streamed to the graphics device.
VertexDeclaration	Represents a vertex declaration.
VertexShader	Encapsulates the functionality of a vertex shader.
VertexStream	Represents a vertex stream.
VertexStreamCollection	Collection of VertexStream objects.

Interfaces




Name	Description
IGraphicsDeviceService	Defines a mechanism for retrieving GraphicsDevice objects.

Structures

Name	Description
Color	Represents a color using Red, Green, Blue, and Alpha values.
CompiledEffect	Represents a compiled Effect .
CompiledShader	Represents a compiled shader.
CompilerMacro	Represents a compiler macro.
DisplayMode	Describes the display mode.
DisplayModeCollection	Manipulates a collection of DisplayMode structures.
GraphicsDeviceCapabilities.AddressCaps	Represents the texture addressing capabilities for Texture structures.
GraphicsDeviceCapabilities.BlendCaps	Represents the supported blend capabilities.
GraphicsDeviceCapabilities.CompareCaps	Represents comparison capabilities.
GraphicsDeviceCapabilities.CursorCaps	Represents hardware support for cursors.
GraphicsDeviceCapabilities.DeclarationTypeCaps	Represents data types contained in a vertex declaration.

 GraphicsDeviceCapabilities.DeviceCaps	Represents device-specific capabilities.
 GraphicsDeviceCapabilities.DriverCaps	Represents driver-specific capabilities.
 GraphicsDeviceCapabilities.FilterCaps	Represents texture filter capabilities.
 GraphicsDeviceCapabilities.LineCaps	Represents the capabilities for line-drawing primitives.
 GraphicsDeviceCapabilities.PixelShaderCaps	Represents pixel shader capabilities.
 GraphicsDeviceCapabilities.PrimitiveCaps	Represents driver primitive capabilities.
 GraphicsDeviceCapabilities.RasterCaps	Represents raster-drawing capabilities.
 GraphicsDeviceCapabilities.ShadingCaps	Represents shading operations capabilities
 GraphicsDeviceCapabilities.StencilCaps	Represents driver stencil capabilities.
 GraphicsDeviceCapabilities.TextureCaps	Represents miscellaneous texture-mapping capabilities
 GraphicsDeviceCapabilities.VertexFormatCaps	Represents flexible vertex format capabilities.
 GraphicsDeviceCapabilities.VertexProcessingCaps	Represents vertex processing capabilities.
 GraphicsDeviceCapabilities.VertexShaderCaps	Represents vertex shader version 2_0 extended capabilities.
 GraphicsDeviceCreationParameters	Describes the creation parameters for a device.
 ModelBoneCollection.Enumerator	Provides the ability to iterate through the bones in an ModelBoneCollection .
 ModelEffectCollection.Enumerator	Provides the ability to iterate through the bones in an ModelEffectCollection .
 ModelMeshCollection.Enumerator	Provides the ability to iterate through the bones in an ModelMeshCollection .
 ModelMeshPartCollection.Enumerator	Provides the ability to iterate through the bones in an ModelMeshPartCollection .
 RasterStatus	Describes the raster status.
 ShaderSemantic	Semantics map a parameter to vertex or pixel shader registers. They can also be optional descriptive strings attached to non-register parameters.
 TextureCreationParameters	Describes the parameters to use when initializing a new instance of a texture.
 TextureInformation	Encapsulates information describing texture resources.
 VertexElement	Defines input vertex data to the pipeline.
 VertexPositionColor	Describes a custom vertex format structure that contains position and color information.
 VertexPositionColorTexture	Describes a custom vertex format structure that contains position, color, and one set of texture coordinates.
 VertexPositionNormalTexture	Describes a custom vertex format structure that contains position, normal data, and one set of texture coordinates.
 VertexPositionTexture	Describes a custom vertex format structure that contains position and one set of texture coordinates.
 Viewport	Defines the window dimensions of a render-target surface onto which a 3D volume projects.

Enumerations

Name	Description
 Blend	Defines color blending factors.
 BlendFunction	Defines how to combine a source color with the destination color already on the render target for color blending.
 BufferUsage	Specifies special usage of the buffer contents.

 ClearOptions	Specifies the buffer to use when calling Clear .
 ColorWriteChannels	Defines the color channels that can be chosen for a per-channel write to a render target color buffer.
 CompareFunction	Defines comparison functions that can be chosen for alpha, stencil, or depth-buffer tests.
 CompilerIncludeHandlerType	Identifies an include file as a local or system resource.
 CompilerOptions	Defines optimization options that may be chosen for shader and effect code compilation.
 CubeMapFace	Defines the faces of a cube map in the TextureCube class type.
 CullMode	Defines winding orders that may be used to identify back faces for culling.
 DepthFormat	Defines the format of data in a depth buffer.
 DeviceType	Specifies the type of device driver.
 EffectParameterClass	Defines classes that can be used for effect parameters or shader constants.
 EffectParameterType	Defines types that can be used for effect parameters or shader constants.
 FillMode	Describes options for filling the vertices and lines that define a primitive.
 FilterOptions	Defines modes describing how to filter an image or mipmap when it is minified or magnified to fit a set of vertices.
 FogMode	Defines constants that describe the fog mode.
 GraphicsDeviceStatus	Describes the status of the device.
 ImageFileFormat	Defines supported image file formats that may be used for textures.
 IndexElementSize	Defines the size of an element of an index buffer.
 MultiSampleType	Defines the levels of full-scene multisampling that the game machine can apply.
 PresentInterval	Defines flags that describe the relationship between the adapter refresh rate and the rate at which Present operations are completed.
 PresentOptions	Defines flags that control the behavior of the back buffer and depth buffer.
 PrimitiveType	Defines how data in a vertex stream is interpreted during a draw call.
 QueryUsages	Defines options for querying device resource formats.
 RenderTargetUsage	Determines how render target data is used once a new render target is set.
 ResourceType	Defines resource types.
 SaveStateMode	Defines options for saving the graphics device state before and after an effect technique is applied.
 SetDataOptions	Describes whether existing buffer data will be overwritten or discarded during a SetData operation.
 ShaderProfile	Defines vertex and pixel shader versions.
 ShaderRegisterSet	Defines the data type of a shader register.
 SpriteBlendMode	The following flags are used to specify sprite blending rendering options to the flags parameter in Begin :
 SpriteEffects	Defines sprite mirroring options.
 SpriteSortMode	Defines sprite sort-rendering options.
 StencilOperation	Defines stencil buffer operations.
 SurfaceFormat	Defines various types of surface formats.
 SwapEffect	Defines how the device front buffer and back buffer are to be swapped when Present is called.
 TextureAddressMode	Defines constants that describe supported texture-addressing modes.
 TextureFilter	Defines how a texture will be filtered as it is minified for each mipmap level.
 TextureUsage	Specifies special usage of the texture data.
 TextureWrapCoordinates	Defines supported wrap coordinates.
 VertexElementFormat	Defines vertex element formats.
 VertexElementMethod	Defines the tessellator processing method for a vertex element.
 VertexElementUsage	Defines usage for vertex elements.

See Also

Tasks

[How To: Draw a Sprite](#)

[How To: Animate a Sprite](#)

[How To: Draw a Masked Sprite over a Background](#)

[How To: Make a Scrolling Background](#)

[How To: Rotate a Sprite](#)

[How To: Rotate a Group of Sprites](#)

[How To: Scale a Sprite](#)

[How To: Tint a Sprite](#)

[How To: Check for Shader Model 2.0 Support](#)

[How To: Use BasicEffect](#)

[How To: Draw Points, Lines, and Other 3D Primitives](#)

[How To: Create and Apply Custom Effects](#)

[How To: Create Custom Texture Effects](#)

Concepts

[2D Graphics Overview](#)

[3D Graphics Overview](#)

[Shader Content Catalog at XNA Creators Club Online](#)

BasicDirectionalLight Class

Provides methods for manipulating the directional lights used by the [BasicEffect](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class BasicDirectionalLight
```

See Also

Reference

[BasicDirectionalLight Members](#)





[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






BasicDirectionalLight Members

The following tables list the members exposed by the BasicDirectionalLight type.



Public Properties

	Name	Description
	DiffuseColor	Gets or sets the diffuse color of this light.
	Direction	Gets or sets light direction.
	Enabled	Enables this light.
	SpecularColor	Gets or sets the specular color of the light.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[BasicDirectionalLight Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicDirectionalLight Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[BasicDirectionalLight Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicDirectionalLight Properties

Public Properties

	Name	Description
	DiffuseColor	Gets or sets the diffuse color of this light.
	Direction	Gets or sets light direction.
	Enabled	Enables this light.
	SpecularColor	Gets or sets the specular color of the light.

See Also

Reference

[BasicDirectionalLight Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicDirectionalLight.DiffuseColor Property

Gets or sets the diffuse color of this light.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 DiffuseColor { get; set; }
```

Property Value

The diffuse color of this light.

See Also

Reference

[BasicDirectionalLight Class](#)

[BasicDirectionalLight Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicDirectionalLight.Direction Property

Gets or sets light direction.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Direction { get; set; }
```

Property Value

Gets or sets the light direction. This value must be a unit vector.

See Also

Reference

[BasicDirectionalLight Class](#)

[BasicDirectionalLight Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicDirectionalLight.Enabled Property

Enables this light.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Enabled { get; set; }
```

Property Value

true to enable this light; **false** otherwise.

See Also

Reference

[BasicDirectionalLight Class](#)

[BasicDirectionalLight Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicDirectionalLight.SpecularColor Property

Gets or sets the specular color of the light.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 SpecularColor { get; set; }
```

Property Value

The specular color of the light.

See Also

Reference

[BasicDirectionalLight Class](#)

[BasicDirectionalLight Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect Class

Represents shader model 1.1 [Effect](#) that has optional support for vertex colors, texturing, and lighting.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class BasicEffect : Effect
```

See Also

Tasks

[How To: Use BasicEffect](#)

[How To: Create and Apply Custom Effects](#)

Reference

[BasicEffect Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






























BasicEffect Members

The following tables list the members exposed by the BasicEffect type.



Public Constructors













Name	Description
 BasicEffect	Overloaded. Initializes a new instance of the BasicEffect class.

Public Properties





Name	Description
 Alpha	Gets or sets the alpha this effect material.
 AmbientLightColor	Gets or sets the ambient light color of this effect.
 Creator	(Inherited from Effect .)
 CurrentTechnique	(Inherited from Effect .)
 DiffuseColor	Gets or sets the diffuse color of this effect material. Value takes 0 to 1.
 DirectionalLight0	Gets the first directional light for this effect.
 DirectionalLight1	Gets the second directional light for this effect.
 DirectionalLight2	Gets the third directional light for this effect.
 EffectPool	(Inherited from Effect .)
 EmissiveColor	Gets or sets the emissive color of the effect material.
 FogColor	Gets or sets the fog color for this effect.
 FogEnabled	Enables fog for this effect.
 FogEnd	Gets or sets the ending distance of fog.
 FogStart	Gets or sets the fog start distance.
 Functions	(Inherited from Effect .)
 GraphicsDevice	(Inherited from Effect .)
 IsDisposed	(Inherited from Effect .)
 LightingEnabled	Enables lighting for this effect.
 Parameters	(Inherited from Effect .)
 PreferPerPixelLighting	Gets or sets a value indicating that per-pixel lighting should be used if it is available for the current adapter. Per-pixel lighting is available if a graphics adapter supports Pixel Shader Model 2.0.
 Projection	Gets or sets the projection matrix.
 SpecularColor	Gets or sets the specular color of this effect material.
 SpecularPower	Gets or sets the specular power of this effect material.
 Techniques	(Inherited from Effect .)
 Texture	Gets or sets a texture to be applied by this effect.
 TextureEnabled	Enables textures for this effect.
 VertexColorEnabled	Enables use vertex colors for this effect.
 View	Gets or sets the view matrix.
 World	Gets or sets the world matrix.

Public Methods




Name	Description
 Begin	(Inherited from Effect .)
 Clone	Creates a clone of an effect.

 CommitChanges	(Inherited from Effect .)
 CompileEffectFromFile	(Inherited from Effect .)
 CompileEffectFromSource	(Inherited from Effect .)
 Disassemble	(Inherited from Effect .)
 Dispose	(Inherited from Effect .)
 EnableDefaultLighting	Enables default lighting for this effect.
 End	(Inherited from Effect .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from Effect .)
	raise_Lost	(Inherited from Effect .)
	raise_Reset	(Inherited from Effect .)

Public Events

	Name	Description
	Disposing	(Inherited from Effect .)
	Lost	(Inherited from Effect .)
	Reset	(Inherited from Effect .)

See Also

Reference

[BasicEffect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicEffect Constructor

Initializes a new instance of the [BasicEffect](#) class.

Overload List

Name	Description
BasicEffect (GraphicsDevice, BasicEffect)	Initializes a new instance of the BasicEffect class.
BasicEffect (GraphicsDevice, EffectPool)	Initializes a new instance of the BasicEffect class.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicEffect Constructor (GraphicsDevice, BasicEffect)

Initializes a new instance of the [BasicEffect](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected BasicEffect (  
    GraphicsDevice device,  
    BasicEffect clone  
)
```

Parameters

device

The graphics device that will create the effect.

clone

The effect to clone.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect Constructor (GraphicsDevice, EffectPool)

Initializes a new instance of the [BasicEffect](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BasicEffect (  
    GraphicsDevice device,  
    EffectPool effectPool  
)
```

Parameters

device

The graphics device that will create the effect.

effectPool

Specifies a pool of resources to share between effects.

See Also

Reference

[BasicEffect Class](#)















[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect Methods

Public Methods

	Name	Description
	Begin	(Inherited from Effect .)
	Clone	Creates a clone of an effect.
	CommitChanges	(Inherited from Effect .)
	CompileEffectFromFile	(Inherited from Effect .)
	CompileEffectFromSource	(Inherited from Effect .)
	Disassemble	(Inherited from Effect .)
	Dispose	(Inherited from Effect .)
	EnableDefaultLighting	Enables default lighting for this effect.
	End	(Inherited from Effect .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from Effect .)
	raise_Lost	(Inherited from Effect .)
	raise_Reset	(Inherited from Effect .)

See Also

Reference

[BasicEffect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicEffect.Clone Method

Creates a clone of an effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override Effect Clone (  
    GraphicsDevice device  
)
```

Parameters

device

The cloned effect.

Return Value

The device associated with the effect.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.EnableDefaultLighting Method

Enables default lighting for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void EnableDefaultLighting ()
```

See Also

Reference

[BasicEffect Class](#)






























[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect Properties

Public Properties

	Name	Description
	Alpha	Gets or sets the alpha this effect material.
	AmbientLightColor	Gets or sets the ambient light color of this effect.
	Creator	(Inherited from Effect .)
	CurrentTechnique	(Inherited from Effect .)
	DiffuseColor	Gets or sets the diffuse color of this effect material. Value takes 0 to 1.
	DirectionalLight0	Gets the first directional light for this effect.
	DirectionalLight1	Gets the second directional light for this effect.
	DirectionalLight2	Gets the third directional light for this effect.
	EffectPool	(Inherited from Effect .)
	EmissiveColor	Gets or sets the emissive color of the effect material.
	FogColor	Gets or sets the fog color for this effect.
	FogEnabled	Enables fog for this effect.
	FogEnd	Gets or sets the ending distance of fog.
	FogStart	Gets or sets the fog start distance.
	Functions	(Inherited from Effect .)
	GraphicsDevice	(Inherited from Effect .)
	IsDisposed	(Inherited from Effect .)
	LightingEnabled	Enables lighting for this effect.
	Parameters	(Inherited from Effect .)
	PreferPerPixelLighting	Gets or sets a value indicating that per-pixel lighting should be used if it is available for the current adapter. Per-pixel lighting is available if a graphics adapter supports Pixel Shader Model 2.0.
	Projection	Gets or sets the projection matrix.
	SpecularColor	Gets or sets the specular color of this effect material.
	SpecularPower	Gets or sets the specular power of this effect material.
	Techniques	(Inherited from Effect .)
	Texture	Gets or sets a texture to be applied by this effect.
	TextureEnabled	Enables textures for this effect.
	VertexColorEnabled	Enables use vertex colors for this effect.
	View	Gets or sets the view matrix.
	World	Gets or sets the world matrix.

See Also

Reference

[BasicEffect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

BasicEffect.Alpha Property

Gets or sets the alpha this effect material.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Alpha { get; set; }
```

Property Value

The alpha of this effect material.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.AmbientLightColor Property

Gets or sets the ambient light color of this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 AmbientLightColor { get; set; }
```

Property Value

The ambient light color of this effect. Valid values are between 0 and 1.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.DiffuseColor Property

Gets or sets the diffuse color of this effect material. Value takes 0 to 1.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 DiffuseColor { get; set; }
```

Property Value

The diffuse color of this effect material. Valid values are between 0 and 1.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.Directionallight0 Property

Gets the first directional light for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BasicDirectionallight Directionallight0 { get; }
```

Property Value

A directional light for this effect.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.Directionallight1 Property

Gets the second directional light for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BasicDirectionallight Directionallight1 { get; }
```

Property Value

A directional light for this effect.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.Directionallight2 Property

Gets the third directional light for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BasicDirectionallight Directionallight2 { get; }
```

Property Value

A directional light for this effect.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.EmissiveColor Property

Gets or sets the emissive color of the effect material.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 EmissiveColor { get; set; }
```

Property Value

The emissive color of the effect material. Valid values are between 0 and 1.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.FogColor Property

Gets or sets the fog color for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 FogColor { get; set; }
```

Property Value

The fog color for this effect.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.FogEnabled Property

Enables fog for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool FogEnabled { get; set; }
```

Property Value

true to enable fog; **false** otherwise.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.FogEnd Property

Gets or sets the ending distance of fog.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float FogEnd { get; set; }
```

Property Value

Fog end distance specified as a positive value.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.FogStart Property

Gets or sets the fog start distance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float FogStart { get; set; }
```

Property Value

Fog start distance specified as a positive value. This distance is expressed in world space coordinates.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.LightingEnabled Property

Enables lighting for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool LightingEnabled { get; set; }
```

Property Value

true to enable lighting; **false** otherwise.

Remarks The lighting for [BasicEffect](#) is per-vertex and uses three directional lights: [DirectionalLight0](#), [DirectionalLight1](#), and [DirectionalLight2](#).

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.PreferPerPixelLighting Property

Gets or sets a value indicating that per-pixel lighting should be used if it is available for the current adapter. Per-pixel lighting is available if a graphics adapter supports Pixel Shader Model 2.0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool PreferPerPixelLighting { get; set; }
```

Property Value

true to use per-pixel lighting if it is available; **false** to disable per-pixel lighting. The default value is **true**. When **PreferPerPixelLighting** is **true**, if the graphics adapter does not support a minimum of Pixel Shader Model 2.0, [BasicEffect](#) will automatically fall-back to per-vertex lighting. When **PreferPerPixelLighting** is **false**, per-vertex lighting is used regardless of whether per-pixel lighting is supported by the graphics adapter.

Remarks

There are two common types of real-time lighting used in games. Per-vertex lighting is where lighting contributions are determined at each vertex and then the results are interpolated across the triangle. Depending on how tessellated the mesh is, this can result in severe artifacts. Per-pixel lighting interpolates the per-vertex normal across the triangle, but calculates the contribution at each pixel, which results in much smoother lighting, particularly when a specular highlight falls between vertices.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.Projection Property

Gets or sets the projection matrix. Use this matrix to change how a 3D image is converted to a 2D image that is rendered to the computer screen.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix Projection { get; set; }
```

Property Value

The projection matrix.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Tasks

[Tutorial 1: Displaying a 3D Model on the Screen](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.SpecularColor Property

Gets or sets the specular color of this effect material.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 SpecularColor { get; set; }
```

Property Value

The specular color of this effect material. Valid values are between 0 and 1.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.SpecularPower Property

Gets or sets the specular power of this effect material.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float SpecularPower { get; set; }
```

Property Value

The specular power of this effect material.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.Texture Property

Gets or sets a texture to be applied by this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D Texture { get; set; }
```

Property Value

Texture to be applied by this effect.

Remarks If the [TextureFilter](#) sampler state is not set, the default filter for **BasicEffect** is [TextureFilter.Linear](#).

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.TextureEnabled Property

Enables textures for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool TextureEnabled { get; set; }
```

Property Value

true to enable textures; **false** otherwise.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.VertexColorEnabled Property

Enables use vertex colors for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool VertexColorEnabled { get; set; }
```

Property Value

true to enable vertex colors; **false** otherwise.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.View Property

Gets or sets the view matrix. Use this matrix to change the position and direction of the camera.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix View { get; set; }
```

Property Value

The view matrix.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Tasks

[Tutorial 1: Displaying a 3D Model on the Screen](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect.World Property

Gets or sets the world matrix. Use this matrix to change the position of the model, using world coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix World { get; set; }
```

Property Value

The world matrix.

See Also

Reference

[BasicEffect Class](#)

[BasicEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Tasks

[Tutorial 1: Displaying a 3D Model on the Screen](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BasicEffect Events

Public Events

	Name	Description
	Disposing	(Inherited from Effect .)
	Lost	(Inherited from Effect .)
	Reset	(Inherited from Effect .)

See Also

Reference

[BasicEffect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Blend Enumeration

Defines color blending factors.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum Blend
```

Members

Member name	Description
Zero	Each component of the color is multiplied by (0, 0, 0, 0).
One	Each component of the color is multiplied by (1, 1, 1, 1).
SourceColor	Each component of the color is multiplied by the source color. This can be represented as (R_S, G_S, B_S, A_S) , where R, G, B, and A respectively stand for the red, green, blue, and alpha source values.
InverseSourceColor	Each component of the color is multiplied by the inverse of the source color. This can be represented as $(1 - R_S, 1 - G_S, 1 - B_S, 1 - A_S)$ where R, G, B, and A respectively stand for the red, green, blue, and alpha destination values.
SourceAlpha	Each component of the color is multiplied by the alpha value of the source. This can be represented as (A_S, A_S, A_S, A_S) , where A_S is the alpha source value.
InverseSourceAlpha	Each component of the color is multiplied by the inverse of the alpha value of the source. This can be represented as $(1 - A_S, 1 - A_S, 1 - A_S, 1 - A_S)$, where A_S is the alpha destination value.
DestinationAlpha	Each component of the color is multiplied by the alpha value of the destination. This can be represented as (A_D, A_D, A_D, A_D) , where A_D is the destination alpha value.
InverseDestinationAlpha	Each component of the color is multiplied by the inverse of the alpha value of the destination. This can be represented as $(1 - A_D, 1 - A_D, 1 - A_D, 1 - A_D)$, where A_D is the alpha destination value.
DestinationColor	Each component color is multiplied by the destination color. This can be represented as (R_D, G_D, B_D, A_D) , where R, G, B, and A respectively stand for red, green, blue, and alpha destination values.
InverseDestinationColor	Each component of the color is multiplied by the inverse of the destination color. This can be represented as $(1 - R_D, 1 - G_D, 1 - B_D, 1 - A_D)$, where $R_D, G_D, B_D,$ and A_D respectively stand for the red, green, blue, and alpha destination values.
SourceAlphaSaturation	Each component of the color is multiplied by either the alpha of the source color, or the inverse of the alpha of the source color, whichever is greater. This can be represented as $(f, f, f, 1)$, where $f = \min(A, 1 - A_D)$.
BothInverseSourceAlpha	(Win32 only) Each component of the source color is multiplied by the inverse of the alpha of the source color, and each component of the destination color is multiplied by the alpha of the source color. This can be represented as $(1 - A_S, 1 - A_S, 1 - A_S, 1 - A_S)$, with a destination blend factor of (A_S, A_S, A_S, A_S) ; the destination blend selection is overridden. This blend mode is supported only for the SourceBlend render state.
BlendFactor	Each component of the color is multiplied by a constant set in BlendFactor .

InverseBlendFactor	Each component of the color is multiplied by the inverse of a constant set in BlendFactor . This blend mode is supported only if SupportsBlendFactor is true in the SourceBlendCapabilities or DestinationBlendCapabilities properties.
BothSourceAlpha	This mode is obsolete. The same effect can be achieved by setting the source and destination blend factors to SourceAlpha and InverseSourceAlpha in separate calls.

Remarks

The alpha value of a color controls its transparency. Enabling alpha blending with [AlphaBlendEnable](#) allows colors, materials, and textures on a surface to be blended with transparency onto another surface.

Alpha blending is determined by the [SourceBlend](#) and [DestinationBlend](#) properties. With the default [BlendFunction](#), the source pixel times the [SourceBlend](#) is added to the destination pixel (the pixel on the back buffer) times the [DestinationBlend](#) to produce a color that is a blend of the two pixel colors. The [BlendFunction](#) determines the operation used on the outcome of the source blend and destination blend components.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps](#) Structure

[RenderState.AlphaDestinationBlend](#) Property

[RenderState.AlphaSourceBlend](#) Property

[RenderState.DestinationBlend](#) Property

[RenderState.SourceBlend](#) Property

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista

BlendFunction Enumeration

Defines how to combine a source color with the destination color already on the render target for color blending.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum BlendFunction
```

Members

Member name	Description
Add	The result is the destination added to the source. <i>Result = (Source Color * Source Blend) + (Destination Color * Destination Blend)</i>
Max	The result is the maximum of the source and destination. <i>Result = max((Source Color * Source Blend), (Destination Color * Destination Blend))</i>
Min	The result is the minimum of the source and destination. <i>Result = min((Source Color * Source Blend), (Destination Color * Destination Blend))</i>
ReverseSubtract	The result is the source subtracted from the destination. <i>Result = (Destination Color * Destination Blend) - (Source Color * Source Blend)</i>
Subtract	The result is the destination subtracted from the source. <i>Result = (Source Color * Source Blend) - (Destination Color * Destination Blend)</i>

Remarks

The alpha value of a color controls its transparency. Enabling alpha blending with [AlphaBlendEnable](#) allows colors, materials, and textures on a surface to be blended with transparency onto another surface.

Alpha blending is determined by the [SourceBlend](#) and [DestinationBlend](#) properties. With the default [BlendFunction](#), the source pixel times the [SourceBlend](#) is added to the destination pixel (the pixel on the back buffer) times the [DestinationBlend](#) to produce a color that is a blend of the two pixel colors. The [BlendFunction](#) determines the operation used on the outcome of the source blend and destination blend components.

See Also

Reference

[BlendFunction](#)

[SourceBlend](#)

[DestinationBlend](#)

[SupportsBlendOperation](#)

[AlphaSourceBlend](#)

[AlphaDestinationBlend](#)

[AlphaBlendOperation](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

BufferUsage Enumeration

Specifies special usage of the buffer contents.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum BufferUsage
```

Members

Member name	Description
None	None
Points	Indicates that the vertex or index buffer will be used for drawing point sprites. The buffer is loaded in system memory if software vertex processing is needed to emulate point sprites.
WriteOnly	Indicates that the application only writes to the vertex buffer. If specified, the driver chooses the best memory location for efficient writing and rendering. Attempts to read from a write-only vertex buffer fail.

See Also

Reference

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ClearOptions Enumeration

Specifies the buffer to use when calling [Clear](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum ClearOptions
```

Members

Member name	Description
DepthBuffer	A depth buffer.
Stencil	A stencil buffer.
Target	A render target.

See Also

Reference

[Clear](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ClipPlane Class

Represents the device clipping plane.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ClipPlane
```

See Also

Reference

[ClipPlane Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ClipPlane Members

The following tables list the members exposed by the ClipPlane type.



Public Properties

	Name	Description
	IsEnabled	Indicates whether the clipping plane is enabled.
	Plane	Gets or sets the coefficients of a user-defined clipping plane for the device.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Gets a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ClipPlane Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ClipPlane Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Gets a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ClipPlane Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ClipPlane.ToString Method

Gets a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[ClipPlane Class](#)



[ClipPlane Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ClipPlane Properties

Public Properties

	Name	Description
	IsEnabled	Indicates whether the clipping plane is enabled.
	Plane	Gets or sets the coefficients of a user-defined clipping plane for the device.

See Also

Reference

[ClipPlane Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ClipPlane.IsEnabled Property

Indicates whether the clipping plane is enabled.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsEnabled { get; set; }
```

Property Value

true if the clipping plane is enabled; **false** if disabled.

See Also

Reference

[ClipPlane Class](#)

[ClipPlane Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ClipPlane.Plane Property

Gets or sets the coefficients of a user-defined clipping plane for the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Plane Plane { get; set; }
```

Property Value

User-defined clipping plane.

See Also

Reference

[ClipPlane Class](#)

[ClipPlane Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ClipPlaneCollection Class

Manipulates a collection of [ClipPlane](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ClipPlaneCollection
```

See Also

Reference

[ClipPlaneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








ClipPlaneCollection Members

The following tables list the members exposed by the ClipPlaneCollection type.



Public Properties

	Name	Description
	Item	Gets a specific ClipPlane object using an index value.

Public Methods

	Name	Description
	DisableAll	Disables all clipping planes on the device.
	EnableAll	Enables all clipping planes that have been set.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[ClipPlaneCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ClipPlaneCollection Methods

Public Methods

	Name	Description
	DisableAll	Disables all clipping planes on the device.
	EnableAll	Enables all clipping planes that have been set.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ClipPlaneCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ClipPlaneCollection.DisableAll Method

Disables all clipping planes on the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DisableAll ()
```

See Also

Reference

[ClipPlaneCollection Class](#)

[ClipPlaneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ClipPlaneCollection.EnableAll Method

Enables all clipping planes that have been set.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void EnableAll ()
```

See Also

Reference

[ClipPlaneCollection Class](#)

[ClipPlaneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ClipPlaneCollection Properties

Public Properties

	Name	Description
	Item	Gets a specific ClipPlane object using an index value.

See Also

Reference

[ClipPlaneCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ClipPlaneCollection.Item Property

Gets a specific [ClipPlane](#) object using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ClipPlane this [
    int index
] { get; }
```

Property Value

The [ClipPlane](#) object at index *index*.

Exceptions

Exception type	Condition
InvalidOperationException	<i>index</i> was out of range.

See Also

Reference

[ClipPlaneCollection Class](#)

[ClipPlaneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Color Structure

Represents a color using Red, Green, Blue, and Alpha values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Color : IPackedVector<uint>, IEquatable<Color>
```

See Also

Reference

[Color Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Members

The following tables list the members exposed by the Color type.


















































Public Constructors

	Name	Description
	Color	Overloaded. Initializes a new instance of Color.




Public Properties








	Name	Description
	A	Gets or sets the alpha component value.
	AliceBlue	Gets a system-defined color with the value R:240 G:248 B:255 A:255.
	AntiqueWhite	Gets a system-defined color with the value R:250 G:235 B:215 A:255.
	Aqua	Gets a system-defined color with the value R:0 G:255 B:255 A:255.
	Aquamarine	Gets a system-defined color with the value R:127 G:255 B:212 A:255.
	Azure	Gets a system-defined color with the value R:240 G:255 B:255 A:255.
	B	Gets or sets the blue component value of this Color.
	Beige	Gets a system-defined color with the value R:245 G:245 B:220 A:255.
	Bisque	Gets a system-defined color with the value R:255 G:228 B:196 A:255.
	Black	Gets a system-defined color with the value R:0 G:0 B:0 A:255.
	BlanchedAlmond	Gets a system-defined color with the value R:255 G:235 B:205 A:255.
	Blue	Gets a system-defined color with the value R:0 G:0 B:255 A:255.
	BlueViolet	Gets a system-defined color with the value R:138 G:43 B:226 A:255.
	Brown	Gets a system-defined color with the value R:165 G:42 B:42 A:255.
	BurlyWood	Gets a system-defined color with the value R:222 G:184 B:135 A:255.
	CadetBlue	Gets a system-defined color with the value R:95 G:158 B:160 A:255.
	Chartreuse	Gets a system-defined color with the value R:127 G:255 B:0 A:255.
	Chocolate	Gets a system-defined color with the value R:210 G:105 B:30 A:255.
	Coral	Gets a system-defined color with the value R:255 G:127 B:80 A:255.
	CornflowerBlue	Gets a system-defined color with the value R:100 G:149 B:237 A:255.
	Cornsilk	Gets a system-defined color with the value R:255 G:248 B:220 A:255.
	Crimson	Gets a system-defined color with the value R:220 G:20 B:60 A:255.
	Cyan	Gets a system-defined color with the value R:0 G:255 B:255 A:255.
	DarkBlue	Gets a system-defined color with the value R:0 G:0 B:139 A:255.
	DarkCyan	Gets a system-defined color with the value R:0 G:139 B:139 A:255.
	DarkGoldenrod	Gets a system-defined color with the value R:184 G:134 B:11 A:255.
	DarkGray	Gets a system-defined color with the value R:169 G:169 B:169 A:255.
	DarkGreen	Gets a system-defined color with the value R:0 G:100 B:0 A:255.
	DarkKhaki	Gets a system-defined color with the value R:189 G:183 B:107 A:255.
	DarkMagenta	Gets a system-defined color with the value R:139 G:0 B:139 A:255.
	DarkOliveGreen	Gets a system-defined color with the value R:85 G:107 B:47 A:255.
	DarkOrange	Gets a system-defined color with the value R:255 G:140 B:0 A:255.
	DarkOrchid	Gets a system-defined color with the value R:153 G:50 B:204 A:255.
	DarkRed	Gets a system-defined color with the value R:139 G:0 B:0 A:255.
	DarkSalmon	Gets a system-defined color with the value R:233 G:150 B:122 A:255.
	DarkSeaGreen	Gets a system-defined color with the value R:143 G:188 B:139 A:255.
	DarkSlateBlue	Gets a system-defined color with the value R:72 G:61 B:139 A:255.
	DarkSlateGray	Gets a system-defined color with the value R:47 G:79 B:79 A:255.
	DarkTurquoise	Gets a system-defined color with the value R:0 G:206 B:209 A:255.
	DarkViolet	Gets a system-defined color with the value R:148 G:0 B:211 A:255.
	DeepPink	Gets a system-defined color with the value R:255 G:20 B:147 A:255.
	DeepSkyBlue	Gets a system-defined color with the value R:0 G:191 B:255 A:255.
	DimGray	Gets a system-defined color with the value R:105 G:105 B:105 A:255.
	DodgerBlue	Gets a system-defined color with the value R:30 G:144 B:255 A:255.

 S Firebrick	Gets a system-defined color with the value R:178 G:34 B:34 A:255.
 S FloralWhite	Gets a system-defined color with the value R:255 G:250 B:240 A:255.
 S ForestGreen	Gets a system-defined color with the value R:34 G:139 B:34 A:255.
 S Fuchsia	Gets a system-defined color with the value R:255 G:0 B:255 A:255.
 G	Gets or sets the green component value of this Color .
 S Gainsboro	Gets a system-defined color with the value R:220 G:220 B:220 A:255.
 S GhostWhite	Gets a system-defined color with the value R:248 G:248 B:255 A:255.
 S Gold	Gets a system-defined color with the value R:255 G:215 B:0 A:255.
 S Goldenrod	Gets a system-defined color with the value R:218 G:165 B:32 A:255.
 S Gray	Gets a system-defined color with the value R:128 G:128 B:128 A:255.
 S Green	Gets a system-defined color with the value R:0 G:128 B:0 A:255.
 S GreenYellow	Gets a system-defined color with the value R:173 G:255 B:47 A:255.
 S Honeydew	Gets a system-defined color with the value R:240 G:255 B:240 A:255.
 S HotPink	Gets a system-defined color with the value R:255 G:105 B:180 A:255.
 S IndianRed	Gets a system-defined color with the value R:205 G:92 B:92 A:255.
 S Indigo	Gets a system-defined color with the value R:75 G:0 B:130 A:255.
 S Ivory	Gets a system-defined color with the value R:255 G:255 B:240 A:255.
 S Khaki	Gets a system-defined color with the value R:240 G:230 B:140 A:255.
 S Lavender	Gets a system-defined color with the value R:230 G:230 B:250 A:255.
 S LavenderBlush	Gets a system-defined color with the value R:255 G:240 B:245 A:255.
 S LawnGreen	Gets a system-defined color with the value R:124 G:252 B:0 A:255.
 S LemonChiffon	Gets a system-defined color with the value R:255 G:250 B:205 A:255.
 S LightBlue	Gets a system-defined color with the value R:173 G:216 B:230 A:255.
 S LightCoral	Gets a system-defined color with the value R:240 G:128 B:128 A:255.
 S LightCyan	Gets a system-defined color with the value R:224 G:255 B:255 A:255.
 S LightGoldenrodYellow	Gets a system-defined color with the value R:250 G:250 B:210 A:255.
 S LightGray	Gets a system-defined color with the value R:211 G:211 B:211 A:255.
 S LightGreen	Gets a system-defined color with the value R:144 G:238 B:144 A:255.
 S LightPink	Gets a system-defined color with the value R:255 G:182 B:193 A:255.
 S LightSalmon	Gets a system-defined color with the value R:255 G:160 B:122 A:255.
 S LightSeaGreen	Gets a system-defined color with the value R:32 G:178 B:170 A:255.
 S LightSkyBlue	Gets a system-defined color with the value R:135 G:206 B:250 A:255.
 S LightSlateGray	Gets a system-defined color with the value R:119 G:136 B:153 A:255.
 S LightSteelBlue	Gets a system-defined color with the value R:176 G:196 B:222 A:255.
 S LightYellow	Gets a system-defined color with the value R:255 G:255 B:224 A:255.
 S Lime	Gets a system-defined color with the value R:0 G:255 B:0 A:255.
 S LimeGreen	Gets a system-defined color with the value R:50 G:205 B:50 A:255.
 S Linen	Gets a system-defined color with the value R:250 G:240 B:230 A:255.
 S Magenta	Gets a system-defined color with the value R:255 G:0 B:255 A:255.
 S Maroon	Gets a system-defined color with the value R:128 G:0 B:0 A:255.
 S MediumAquamarine	Gets a system-defined color with the value R:102 G:205 B:170 A:255.
 S MediumBlue	Gets a system-defined color with the value R:0 G:0 B:205 A:255.
 S MediumOrchid	Gets a system-defined color with the value R:186 G:85 B:211 A:255.
 S MediumPurple	Gets a system-defined color with the value R:147 G:112 B:219 A:255.
 S MediumSeaGreen	Gets a system-defined color with the value R:60 G:179 B:113 A:255.
 S MediumSlateBlue	Gets a system-defined color with the value R:123 G:104 B:238 A:255.
 S MediumSpringGreen	Gets a system-defined color with the value R:0 G:250 B:154 A:255.
 S MediumTurquoise	Gets a system-defined color with the value R:72 G:209 B:204 A:255.
 S MediumVioletRed	Gets a system-defined color with the value R:199 G:21 B:133 A:255.
S MidnightBlue	Gets a system-defined color with the value R:25 G:25 B:112 A:255.
S MintCream	Gets a system-defined color with the value R:245 G:255 B:250 A:255.
S MistyRose	Gets a system-defined color with the value R:255 G:228 B:225 A:255.
S Moccasin	Gets a system-defined color with the value R:255 G:228 B:181 A:255.
S NavajoWhite	Gets a system-defined color with the value R:255 G:222 B:173 A:255.



 S Navy	Gets a system-defined color R:0 G:0 B:128 A:255.
 S OldLace	Gets a system-defined color with the value R:253 G:245 B:230 A:255.
 S Olive	Gets a system-defined color with the value R:128 G:128 B:0 A:255.
 S OliveDrab	Gets a system-defined color with the value R:107 G:142 B:35 A:255.
 S Orange	Gets a system-defined color with the value R:255 G:165 B:0 A:255.
 S OrangeRed	Gets a system-defined color with the value R:255 G:69 B:0 A:255.
 S Orchid	Gets a system-defined color with the value R:218 G:112 B:214 A:255.
 S PackedValue	Gets the current color as a packed value.
 S PaleGoldenrod	Gets a system-defined color with the value R:238 G:232 B:170 A:255.
 S PaleGreen	Gets a system-defined color with the value R:152 G:251 B:152 A:255.
 S PaleTurquoise	Gets a system-defined color with the value R:175 G:238 B:238 A:255.
 S PaleVioletRed	Gets a system-defined color with the value R:219 G:112 B:147 A:255.
 S PapayaWhip	Gets a system-defined color with the value R:255 G:239 B:213 A:255.
 S PeachPuff	Gets a system-defined color with the value R:255 G:218 B:185 A:255.
 S Peru	Gets a system-defined color with the value R:205 G:133 B:63 A:255.
 S Pink	Gets a system-defined color with the value R:255 G:192 B:203 A:255.
 S Plum	Gets a system-defined color with the value R:221 G:160 B:221 A:255.
 S PowderBlue	Gets a system-defined color with the value R:176 G:224 B:230 A:255.
 S Purple	Gets a system-defined color with the value R:128 G:0 B:128 A:255.
 R	Gets or sets the red component value of this Color .
 S Red	Gets a system-defined color with the value R:255 G:0 B:0 A:255.
 S RosyBrown	Gets a system-defined color with the value R:188 G:143 B:143 A:255.
 S RoyalBlue	Gets a system-defined color with the value R:65 G:105 B:225 A:255.
 S SaddleBrown	Gets a system-defined color with the value R:139 G:69 B:19 A:255.
 S Salmon	Gets a system-defined color with the value R:250 G:128 B:114 A:255.
 S SandyBrown	Gets a system-defined color with the value R:244 G:164 B:96 A:255.
 S SeaGreen	Gets a system-defined color with the value R:46 G:139 B:87 A:255.
 S SeaShell	Gets a system-defined color with the value R:255 G:245 B:238 A:255.
 S Sienna	Gets a system-defined color with the value R:160 G:82 B:45 A:255.
 S Silver	Gets a system-defined color with the value R:192 G:192 B:192 A:255.
 S SkyBlue	Gets a system-defined color with the value R:135 G:206 B:235 A:255.
 S SlateBlue	Gets a system-defined color with the value R:106 G:90 B:205 A:255.
 S SlateGray	Gets a system-defined color with the value R:112 G:128 B:144 A:255.
 S Snow	Gets a system-defined color with the value R:255 G:250 B:250 A:255.
 S SpringGreen	Gets a system-defined color with the value R:0 G:255 B:127 A:255.
 S SteelBlue	Gets a system-defined color with the value R:70 G:130 B:180 A:255.
 S Tan	Gets a system-defined color with the value R:210 G:180 B:140 A:255.
 S Teal	Gets a system-defined color with the value R:0 G:128 B:128 A:255.
 S Thistle	Gets a system-defined color with the value R:216 G:191 B:216 A:255.
 S Tomato	Gets a system-defined color with the value R:255 G:99 B:71 A:255.
 S TransparentBlack	Gets a system-defined color with the value R:0 G:0 B:0 A:0.
 S TransparentWhite	Gets a system-defined color with the value R:255 G:255 B:255 A:0.
 S Turquoise	Gets a system-defined color with the value R:64 G:224 B:208 A:255.
 S Violet	Gets a system-defined color with the value R:238 G:130 B:238 A:255.
 S Wheat	Gets a system-defined color with the value R:245 G:222 B:179 A:255.
 S White	Gets a system-defined color with the value R:255 G:255 B:255 A:255.
 S WhiteSmoke	Gets a system-defined color with the value R:245 G:245 B:245 A:255.
 S Yellow	Gets a system-defined color with the value R:255 G:255 B:0 A:255.
 S YellowGreen	Gets a system-defined color with the value R:154 G:205 B:50 A:255.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)

 Lerp	Linearly interpolates between two colors.
 Op_Equality	Compares two objects to determine whether they are the same.
 Op_Inequality	Compares two objects to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Gets a string representation of this object.
 ToVector3	Returns the current color as a Vector3 .
 ToVector4	Returns the current color as a Vector4 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Returns the packed value of the current color as a Vector4 .

See Also

Reference

[Color Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Color Constructor

Initializes a new instance of [Color](#).

Overload List

Name	Description
Color (Byte, Byte, Byte)	Initializes a new instance of Color .
Color (Byte, Byte, Byte, Byte)	Initializes a new instance of Color .
Color (Color, Byte)	Initializes a new instance of Color .
Color (Color, Single)	Initializes a new instance of Color .
Color (Single, Single, Single)	Initializes a new instance of Color .
Color (Single, Single, Single, Single)	Initializes a new instance of Color .
Color (Vector3)	Initializes a new instance of Color .
Color (Vector4)	Initializes a new instance of Color .

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Color Constructor (Byte, Byte, Byte)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (  
    byte r,  
    byte g,  
    byte b  
)
```

Parameters

r

The red component of a color, between 0 and 255.

g

The green component of a color, between 0 and 255.

b

The blue component of a color, between 0 and 255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Byte, Byte, Byte, Byte)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (
    byte r,
    byte g,
    byte b,
    byte a
)
```

Parameters

r

The red component of a color, between 0 and 255.

g

The green component of a color, between 0 and 255.

b

The blue component of a color, between 0 and 255.

a

The alpha component of a color, between 0 and 255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Color, Byte)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (
    Color rgb,
    byte a
)
```

Parameters

rgb

A [Color](#) specifying the red, green, and blue components of a color.

a

The alpha component of a color, between 0 and 255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Color, Single)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (  
    Color rgb,  
    float a  
)
```

Parameters

rgb

A [Color](#) specifying the red, green, and blue components of a color.

a

The alpha component of a color, between 0 and 1.0.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Single, Single, Single)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (  
    float r,  
    float g,  
    float b  
)
```

Parameters

r

The red component of a color, between 0 and 1.0.

g

The green component of a color, between 0 and 1.0.

b

The blue component of a color, between 0 and 1.0.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Single, Single, Single, Single)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (
    float r,
    float g,
    float b,
    float a
)
```

Parameters

r

The red component of a color, between 0 and 1.0.

g

The green component of a color, between 0 and 1.0.

b

The blue component of a color, between 0 and 1.0.

a

The alpha component of a color, between 0 and 1.0.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Vector3)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (  
    Vector3 vector  
)
```

Parameters

vector

A [Vector3](#) containing the Red, Green, and Blue values defining a color.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Constructor (Vector4)

Initializes a new instance of [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color (  
    Vector4 vector  
)
```

Parameters

vector

A [Vector4](#) containing the Red, Green, Blue, and Alpha values defining a color.

See Also

Reference

[Color Structure](#)














[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 Lerp	Linearly interpolates between two colors.
	 op_Equality	Compares two objects to determine whether they are the same.
	 op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Gets a string representation of this object.
	ToVector3	Returns the current color as a Vector3 .
	ToVector4	Returns the current color as a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Returns the packed value of the current color as a Vector4 .

See Also

Reference

[Color Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Color.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Color.Equals (Color)	Returns a value that indicates whether the current instance is equal to a specified object.
Color.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Color.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Color.Equals Method (Color)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Color other  
)
```

Parameters

other

The [Color](#) to compare with the current [Color](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [Color](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Lerp Method

Linearly interpolates between two colors.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Lerp (  
    Color value1,  
    Color value2,  
    float amount  
)
```

Parameters

value1

Source [Color](#).

value2

Source [Color](#).

amount

A value between 0 and 1.0 indicating the weight of **value2**.

Return Value

The interpolated [Color](#).

Remarks

This function linearly interpolates each component of a [Color](#) separately and returns a [Color](#) with the new component values.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Returns the packed value of the current color as a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The packed value of the current color.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Color a,  
    Color b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Color a,  
    Color b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.ToString Method

Gets a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.ToVector3 Method

Returns the current color as a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 ToVector3 ()
```

Return Value

The current color represented as a [Vector3](#).

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.ToVector4 Method

Returns the current color as a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The current color.

See Also

Reference

[Color Structure](#)















[Color Members](#)


























[Microsoft.Xna.Framework.Graphics Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color Properties

Public Properties

	Name	Description
	A	Gets or sets the alpha component value.
	AliceBlue	Gets a system-defined color with the value R:240 G:248 B:255 A:255.
	AntiqueWhite	Gets a system-defined color with the value R:250 G:235 B:215 A:255.
	Aqua	Gets a system-defined color with the value R:0 G:255 B:255 A:255.
	Aquamarine	Gets a system-defined color with the value R:127 G:255 B:212 A:255.
	Azure	Gets a system-defined color with the value R:240 G:255 B:255 A:255.
	B	Gets or sets the blue component value of this Color .
	Beige	Gets a system-defined color with the value R:245 G:245 B:220 A:255.
	Bisque	Gets a system-defined color with the value R:255 G:228 B:196 A:255.
	Black	Gets a system-defined color with the value R:0 G:0 B:0 A:255.
	BlanchedAlmond	Gets a system-defined color with the value R:255 G:235 B:205 A:255.
	Blue	Gets a system-defined color with the value R:0 G:0 B:255 A:255.
	BlueViolet	Gets a system-defined color with the value R:138 G:43 B:226 A:255.
	Brown	Gets a system-defined color with the value R:165 G:42 B:42 A:255.
	BurlyWood	Gets a system-defined color with the value R:222 G:184 B:135 A:255.
	CadetBlue	Gets a system-defined color with the value R:95 G:158 B:160 A:255.
	Chartreuse	Gets a system-defined color with the value R:127 G:255 B:0 A:255.
	Chocolate	Gets a system-defined color with the value R:210 G:105 B:30 A:255.
	Coral	Gets a system-defined color with the value R:255 G:127 B:80 A:255.
	CornflowerBlue	Gets a system-defined color with the value R:100 G:149 B:237 A:255.
	Cornsilk	Gets a system-defined color with the value R:255 G:248 B:220 A:255.
	Crimson	Gets a system-defined color with the value R:220 G:20 B:60 A:255.
	Cyan	Gets a system-defined color with the value R:0 G:255 B:255 A:255.
	DarkBlue	Gets a system-defined color with the value R:0 G:0 B:139 A:255.
	DarkCyan	Gets a system-defined color with the value R:0 G:139 B:139 A:255.
	DarkGoldenrod	Gets a system-defined color with the value R:184 G:134 B:11 A:255.
	DarkGray	Gets a system-defined color with the value R:169 G:169 B:169 A:255.
	DarkGreen	Gets a system-defined color with the value R:0 G:100 B:0 A:255.
	DarkKhaki	Gets a system-defined color with the value R:189 G:183 B:107 A:255.
	DarkMagenta	Gets a system-defined color with the value R:139 G:0 B:139 A:255.
	DarkOliveGreen	Gets a system-defined color with the value R:85 G:107 B:47 A:255.
	DarkOrange	Gets a system-defined color with the value R:255 G:140 B:0 A:255.
	DarkOrchid	Gets a system-defined color with the value R:153 G:50 B:204 A:255.
	DarkRed	Gets a system-defined color with the value R:139 G:0 B:0 A:255.
	DarkSalmon	Gets a system-defined color with the value R:233 G:150 B:122 A:255.
	DarkSeaGreen	Gets a system-defined color with the value R:143 G:188 B:139 A:255.
	DarkSlateBlue	Gets a system-defined color with the value R:72 G:61 B:139 A:255.
	DarkSlateGray	Gets a system-defined color with the value R:47 G:79 B:79 A:255.
	DarkTurquoise	Gets a system-defined color with the value R:0 G:206 B:209 A:255.
	DarkViolet	Gets a system-defined color with the value R:148 G:0 B:211 A:255.
	DeepPink	Gets a system-defined color with the value R:255 G:20 B:147 A:255.
	DeepSkyBlue	Gets a system-defined color with the value R:0 G:191 B:255 A:255.
	DimGray	Gets a system-defined color with the value R:105 G:105 B:105 A:255.
	DodgerBlue	Gets a system-defined color with the value R:30 G:144 B:255 A:255.
	Firebrick	Gets a system-defined color with the value R:178 G:34 B:34 A:255.
	FloralWhite	Gets a system-defined color with the value R:255 G:250 B:240 A:255.
	ForestGreen	Gets a system-defined color with the value R:34 G:139 B:34 A:255.
	Fuchsia	Gets a system-defined color with the value R:255 G:0 B:255 A:255.
	G	Gets or sets the green component value of this Color .

 S Gainsboro	Gets a system-defined color with the value R:220 G:220 B:220 A:255.
 S GhostWhite	Gets a system-defined color with the value R:248 G:248 B:255 A:255.
 S Gold	Gets a system-defined color with the value R:255 G:215 B:0 A:255.
 S Goldenrod	Gets a system-defined color with the value R:218 G:165 B:32 A:255.
 S Gray	Gets a system-defined color with the value R:128 G:128 B:128 A:255.
 S Green	Gets a system-defined color with the value R:0 G:128 B:0 A:255.
 S GreenYellow	Gets a system-defined color with the value R:173 G:255 B:47 A:255.
 S Honeydew	Gets a system-defined color with the value R:240 G:255 B:240 A:255.
 S HotPink	Gets a system-defined color with the value R:255 G:105 B:180 A:255.
 S IndianRed	Gets a system-defined color with the value R:205 G:92 B:92 A:255.
 S Indigo	Gets a system-defined color with the value R:75 G:0 B:130 A:255.
 S Ivory	Gets a system-defined color with the value R:255 G:255 B:240 A:255.
 S Khaki	Gets a system-defined color with the value R:240 G:230 B:140 A:255.
 S Lavender	Gets a system-defined color with the value R:230 G:230 B:250 A:255.
 S LavenderBlush	Gets a system-defined color with the value R:255 G:240 B:245 A:255.
 S LawnGreen	Gets a system-defined color with the value R:124 G:252 B:0 A:255.
 S LemonChiffon	Gets a system-defined color with the value R:255 G:250 B:205 A:255.
 S LightBlue	Gets a system-defined color with the value R:173 G:216 B:230 A:255.
 S LightCoral	Gets a system-defined color with the value R:240 G:128 B:128 A:255.
 S LightCyan	Gets a system-defined color with the value R:224 G:255 B:255 A:255.
 S LightGoldenrodYellow	Gets a system-defined color with the value R:250 G:250 B:210 A:255.
 S LightGray	Gets a system-defined color with the value R:211 G:211 B:211 A:255.
 S LightGreen	Gets a system-defined color with the value R:144 G:238 B:144 A:255.
 S LightPink	Gets a system-defined color with the value R:255 G:182 B:193 A:255.
 S LightSalmon	Gets a system-defined color with the value R:255 G:160 B:122 A:255.
 S LightSeaGreen	Gets a system-defined color with the value R:32 G:178 B:170 A:255.
 S LightSkyBlue	Gets a system-defined color with the value R:135 G:206 B:250 A:255.
 S LightSlateGray	Gets a system-defined color with the value R:119 G:136 B:153 A:255.
 S LightSteelBlue	Gets a system-defined color with the value R:176 G:196 B:222 A:255.
 S LightYellow	Gets a system-defined color with the value R:255 G:255 B:224 A:255.
 S Lime	Gets a system-defined color with the value R:0 G:255 B:0 A:255.
 S LimeGreen	Gets a system-defined color with the value R:50 G:205 B:50 A:255.
 S Linen	Gets a system-defined color with the value R:250 G:240 B:230 A:255.
 S Magenta	Gets a system-defined color with the value R:255 G:0 B:255 A:255.
 S Maroon	Gets a system-defined color with the value R:128 G:0 B:0 A:255.
 S MediumAquamarine	Gets a system-defined color with the value R:102 G:205 B:170 A:255.
 S MediumBlue	Gets a system-defined color with the value R:0 G:0 B:205 A:255.
 S MediumOrchid	Gets a system-defined color with the value R:186 G:85 B:211 A:255.
 S MediumPurple	Gets a system-defined color with the value R:147 G:112 B:219 A:255.
 S MediumSeaGreen	Gets a system-defined color with the value R:60 G:179 B:113 A:255.
 S MediumSlateBlue	Gets a system-defined color with the value R:123 G:104 B:238 A:255.
 S MediumSpringGreen	Gets a system-defined color with the value R:0 G:250 B:154 A:255.
 S MediumTurquoise	Gets a system-defined color with the value R:72 G:209 B:204 A:255.
 S MediumVioletRed	Gets a system-defined color with the value R:199 G:21 B:133 A:255.
 S MidnightBlue	Gets a system-defined color with the value R:25 G:25 B:112 A:255.
 S MintCream	Gets a system-defined color with the value R:245 G:255 B:250 A:255.
 S MistyRose	Gets a system-defined color with the value R:255 G:228 B:225 A:255.
 S Moccasin	Gets a system-defined color with the value R:255 G:228 B:181 A:255.
 S NavajoWhite	Gets a system-defined color with the value R:255 G:222 B:173 A:255.
S Navy	Gets a system-defined color R:0 G:0 B:128 A:255.
S OldLace	Gets a system-defined color with the value R:253 G:245 B:230 A:255.
S Olive	Gets a system-defined color with the value R:128 G:128 B:0 A:255.
S OliveDrab	Gets a system-defined color with the value R:107 G:142 B:35 A:255.
S Orange	Gets a system-defined color with the value R:255 G:165 B:0 A:255.

 OrangeRed	Gets a system-defined color with the value R:255 G:69 B:0 A:255.
 Orchid	Gets a system-defined color with the value R:218 G:112 B:214 A:255.
 PackedValue	Gets the current color as a packed value.
 PaleGoldenrod	Gets a system-defined color with the value R:238 G:232 B:170 A:255.
 PaleGreen	Gets a system-defined color with the value R:152 G:251 B:152 A:255.
 PaleTurquoise	Gets a system-defined color with the value R:175 G:238 B:238 A:255.
 PaleVioletRed	Gets a system-defined color with the value R:219 G:112 B:147 A:255.
 PapayaWhip	Gets a system-defined color with the value R:255 G:239 B:213 A:255.
 PeachPuff	Gets a system-defined color with the value R:255 G:218 B:185 A:255.
 Peru	Gets a system-defined color with the value R:205 G:133 B:63 A:255.
 Pink	Gets a system-defined color with the value R:255 G:192 B:203 A:255.
 Plum	Gets a system-defined color with the value R:221 G:160 B:221 A:255.
 PowderBlue	Gets a system-defined color with the value R:176 G:224 B:230 A:255.
 Purple	Gets a system-defined color with the value R:128 G:0 B:128 A:255.
 R	Gets or sets the red component value of this Color .
 Red	Gets a system-defined color with the value R:255 G:0 B:0 A:255.
 RosyBrown	Gets a system-defined color with the value R:188 G:143 B:143 A:255.
 RoyalBlue	Gets a system-defined color with the value R:65 G:105 B:225 A:255.
 SaddleBrown	Gets a system-defined color with the value R:139 G:69 B:19 A:255.
 Salmon	Gets a system-defined color with the value R:250 G:128 B:114 A:255.
 SandyBrown	Gets a system-defined color with the value R:244 G:164 B:96 A:255.
 SeaGreen	Gets a system-defined color with the value R:46 G:139 B:87 A:255.
 SeaShell	Gets a system-defined color with the value R:255 G:245 B:238 A:255.
 Sienna	Gets a system-defined color with the value R:160 G:82 B:45 A:255.
 Silver	Gets a system-defined color with the value R:192 G:192 B:192 A:255.
 SkyBlue	Gets a system-defined color with the value R:135 G:206 B:235 A:255.
 SlateBlue	Gets a system-defined color with the value R:106 G:90 B:205 A:255.
 SlateGray	Gets a system-defined color with the value R:112 G:128 B:144 A:255.
 Snow	Gets a system-defined color with the value R:255 G:250 B:250 A:255.
 SpringGreen	Gets a system-defined color with the value R:0 G:255 B:127 A:255.
 SteelBlue	Gets a system-defined color with the value R:70 G:130 B:180 A:255.
 Tan	Gets a system-defined color with the value R:210 G:180 B:140 A:255.
 Teal	Gets a system-defined color with the value R:0 G:128 B:128 A:255.
 Thistle	Gets a system-defined color with the value R:216 G:191 B:216 A:255.
 Tomato	Gets a system-defined color with the value R:255 G:99 B:71 A:255.
 TransparentBlack	Gets a system-defined color with the value R:0 G:0 B:0 A:0.
 TransparentWhite	Gets a system-defined color with the value R:255 G:255 B:255 A:0.
 Turquoise	Gets a system-defined color with the value R:64 G:224 B:208 A:255.
 Violet	Gets a system-defined color with the value R:238 G:130 B:238 A:255.
 Wheat	Gets a system-defined color with the value R:245 G:222 B:179 A:255.
 White	Gets a system-defined color with the value R:255 G:255 B:255 A:255.
 WhiteSmoke	Gets a system-defined color with the value R:245 G:245 B:245 A:255.
 Yellow	Gets a system-defined color with the value R:255 G:255 B:0 A:255.
 YellowGreen	Gets a system-defined color with the value R:154 G:205 B:50 A:255.

See Also

Reference

[Color Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Color.A Property

Gets or sets the alpha component value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte A { get; set; }
```

Property Value

The alpha component value of this [Color](#).

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.AliceBlue Property

Gets a system-defined color with the value R:240 G:248 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color AliceBlue { get; }
```

Property Value

A system-defined color with the value R:240 G:248 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.AntiqueWhite Property

Gets a system-defined color with the value R:250 G:235 B:215 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color AntiqueWhite { get; }
```

Property Value

A system-defined color with the value R:250 G:235 B:215 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Aqua Property

Gets a system-defined color with the value R:0 G:255 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Aqua { get; }
```

Property Value

A system-defined color with the value R:0 G:255 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Aquamarine Property

Gets a system-defined color with the value R:127 G:255 B:212 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Aquamarine { get; }
```

Property Value

A system-defined color with the value R:127 G:255 B:212 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Azure Property

Gets a system-defined color with the value R:240 G:255 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Azure { get; }
```

Property Value

A system-defined color with the value R:240 G:255 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.B Property

Gets or sets the blue component value of this [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte B { get; set; }
```

Property Value

The blue component value of this [Color](#).

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Beige Property

Gets a system-defined color with the value R:245 G:245 B:220 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Beige { get; }
```

Property Value

A system-defined color with the value R:245 G:245 B:220 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Bisque Property

Gets a system-defined color with the value R:255 G:228 B:196 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Bisque { get; }
```

Property Value

A system-defined color with the value R:255 G:228 B:196 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Black Property

Gets a system-defined color with the value R:0 G:0 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Black { get; }
```

Property Value

A system-defined color with the value R:0 G:0 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.BlachedAlmond Property

Gets a system-defined color with the value R:255 G:235 B:205 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color BlachedAlmond { get; }
```

Property Value

A system-defined color with the value R:255 G:235 B:205 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Blue Property

Gets a system-defined color with the value R:0 G:0 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Blue { get; }
```

Property Value

A system-defined color with the value R:0 G:0 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.BlueViolet Property

Gets a system-defined color with the value R:138 G:43 B:226 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color BlueViolet { get; }
```

Property Value

A system-defined color with the value R:138 G:43 B:226 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Brown Property

Gets a system-defined color with the value R:165 G:42 B:42 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Brown { get; }
```

Property Value

A system-defined color with the value R:165 G:42 B:42 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.BurlyWood Property

Gets a system-defined color with the value R:222 G:184 B:135 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color BurlyWood { get; }
```

Property Value

A system-defined color with the value R:222 G:184 B:135 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.CadetBlue Property

Gets a system-defined color with the value R:95 G:158 B:160 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color CadetBlue { get; }
```

Property Value

A system-defined color with the value R:95 G:158 B:160 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Chartreuse Property

Gets a system-defined color with the value R:127 G:255 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Chartreuse { get; }
```

Property Value

A system-defined color with the value R:127 G:255 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Chocolate Property

Gets a system-defined color with the value R:210 G:105 B:30 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Chocolate { get; }
```

Property Value

A system-defined color with the value R:210 G:105 B:30 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Coral Property

Gets a system-defined color with the value R:255 G:127 B:80 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Coral { get; }
```

Property Value

A system-defined color with the value R:255 G:127 B:80 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.CornflowerBlue Property

Gets a system-defined color with the value R:100 G:149 B:237 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color CornflowerBlue { get; }
```

Property Value

A system-defined color with the value R:100 G:149 B:237 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Cornsilk Property

Gets a system-defined color with the value R:255 G:248 B:220 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Cornsilk { get; }
```

Property Value

A system-defined color with the value R:255 G:248 B:220 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Crimson Property

Gets a system-defined color with the value R:220 G:20 B:60 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Crimson { get; }
```

Property Value

A system-defined color with the value R:220 G:20 B:60 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Cyan Property

Gets a system-defined color with the value R:0 G:255 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Cyan { get; }
```

Property Value

A system-defined color with the value R:0 G:255 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkBlue Property

Gets a system-defined color with the value R:0 G:0 B:139 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkBlue { get; }
```

Property Value

A system-defined color with the value R:0 G:0 B:139 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkCyan Property

Gets a system-defined color with the value R:0 G:139 B:139 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkCyan { get; }
```

Property Value

A system-defined color with the value R:0 G:139 B:139 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkGoldenrod Property

Gets a system-defined color with the value R:184 G:134 B:11 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkGoldenrod { get; }
```

Property Value

A system-defined color with the value R:184 G:134 B:11 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkGray Property

Gets a system-defined color with the value R:169 G:169 B:169 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkGray { get; }
```

Property Value

A system-defined color with the value R:169 G:169 B:169 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkGreen Property

Gets a system-defined color with the value R:0 G:100 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkGreen { get; }
```

Property Value

A system-defined color with the value R:0 G:100 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkKhaki Property

Gets a system-defined color with the value R:189 G:183 B:107 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkKhaki { get; }
```

Property Value

A system-defined color with the value R:189 G:183 B:107 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkMagenta Property

Gets a system-defined color with the value R:139 G:0 B:139 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkMagenta { get; }
```

Property Value

A system-defined color with the value R:139 G:0 B:139 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkOliveGreen Property

Gets a system-defined color with the value R:85 G:107 B:47 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkOliveGreen { get; }
```

Property Value

A system-defined color with the value R:85 G:107 B:47 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkOrange Property

Gets a system-defined color with the value R:255 G:140 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkOrange { get; }
```

Property Value

A system-defined color with the value R:255 G:140 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkOrchid Property

Gets a system-defined color with the value R:153 G:50 B:204 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkOrchid { get; }
```

Property Value

A system-defined color with the value R:153 G:50 B:204 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkRed Property

Gets a system-defined color with the value R:139 G:0 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkRed { get; }
```

Property Value

A system-defined color with the value R:139 G:0 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkSalmon Property

Gets a system-defined color with the value R:233 G:150 B:122 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkSalmon { get; }
```

Property Value

A system-defined color with the value R:233 G:150 B:122 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkSeaGreen Property

Gets a system-defined color with the value R:143 G:188 B:139 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkSeaGreen { get; }
```

Property Value

A system-defined color with the value R:143 G:188 B:139 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkSlateBlue Property

Gets a system-defined color with the value R:72 G:61 B:139 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkSlateBlue { get; }
```

Property Value

A system-defined color with the value R:72 G:61 B:139 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkSlateGray Property

Gets a system-defined color with the value R:47 G:79 B:79 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkSlateGray { get; }
```

Property Value

A system-defined color with the value R:47 G:79 B:79 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkTurquoise Property

Gets a system-defined color with the value R:0 G:206 B:209 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkTurquoise { get; }
```

Property Value

A system-defined color with the value R:0 G:206 B:209 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DarkViolet Property

Gets a system-defined color with the value R:148 G:0 B:211 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DarkViolet { get; }
```

Property Value

A system-defined color with the value R:148 G:0 B:211 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DeepPink Property

Gets a system-defined color with the value R:255 G:20 B:147 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DeepPink { get; }
```

Property Value

A system-defined color with the value R:255 G:20 B:147 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DeepSkyBlue Property

Gets a system-defined color with the value R:0 G:191 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DeepSkyBlue { get; }
```

Property Value

A system-defined color with the value R:0 G:191 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DimGray Property

Gets a system-defined color with the value R:105 G:105 B:105 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DimGray { get; }
```

Property Value

A system-defined color with the value R:105 G:105 B:105 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.DodgerBlue Property

Gets a system-defined color with the value R:30 G:144 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color DodgerBlue { get; }
```

Property Value

A system-defined color with the value R:30 G:144 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Firebrick Property

Gets a system-defined color with the value R:178 G:34 B:34 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Firebrick { get; }
```

Property Value

A system-defined color with the value R:178 G:34 B:34 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.FloralWhite Property

Gets a system-defined color with the value R:255 G:250 B:240 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color FloralWhite { get; }
```

Property Value

A system-defined color with the value R:255 G:250 B:240 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.ForestGreen Property

Gets a system-defined color with the value R:34 G:139 B:34 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color ForestGreen { get; }
```

Property Value

A system-defined color with the value R:34 G:139 B:34 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Fuchsia Property

Gets a system-defined color with the value R:255 G:0 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Fuchsia { get; }
```

Property Value

A system-defined color with the value R:255 G:0 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.G Property

Gets or sets the green component value of this [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte G { get; set; }
```

Property Value

The green component value of this [Color](#).

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Gainsboro Property

Gets a system-defined color with the value R:220 G:220 B:220 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Gainsboro { get; }
```

Property Value

A system-defined color with the value R:220 G:220 B:220 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.GhostWhite Property

Gets a system-defined color with the value R:248 G:248 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color GhostWhite { get; }
```

Property Value

A system-defined color with the value R:248 G:248 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Gold Property

Gets a system-defined color with the value R:255 G:215 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Gold { get; }
```

Property Value

A system-defined color with the value R:255 G:215 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Goldenrod Property

Gets a system-defined color with the value R:218 G:165 B:32 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Goldenrod { get; }
```

Property Value

A system-defined color with the value R:218 G:165 B:32 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Gray Property

Gets a system-defined color with the value R:128 G:128 B:128 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Gray { get; }
```

Property Value

A system-defined color with the value R:128 G:128 B:128 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Green Property

Gets a system-defined color with the value R:0 G:128 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Green { get; }
```

Property Value

A system-defined color with the value R:0 G:128 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.GreenYellow Property

Gets a system-defined color with the value R:173 G:255 B:47 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color GreenYellow { get; }
```

Property Value

A system-defined color with the value R:173 G:255 B:47 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Honeydew Property

Gets a system-defined color with the value R:240 G:255 B:240 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Honeydew { get; }
```

Property Value

A system-defined color with the value R:240 G:255 B:240 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.HotPink Property

Gets a system-defined color with the value R:255 G:105 B:180 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color HotPink { get; }
```

Property Value

A system-defined color with the value R:255 G:105 B:180 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.IndianRed Property

Gets a system-defined color with the value R:205 G:92 B:92 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color IndianRed { get; }
```

Property Value

A system-defined color with the value R:205 G:92 B:92 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Indigo Property

Gets a system-defined color with the value R:75 G:0 B:130 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Indigo { get; }
```

Property Value

A system-defined color with the value R:75 G:0 B:130 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Ivory Property

Gets a system-defined color with the value R:255 G:255 B:240 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Ivory { get; }
```

Property Value

A system-defined color with the value R:255 G:255 B:240 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Khaki Property

Gets a system-defined color with the value R:240 G:230 B:140 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Khaki { get; }
```

Property Value

A system-defined color with the value R:240 G:230 B:140 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Lavender Property

Gets a system-defined color with the value R:230 G:230 B:250 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Lavender { get; }
```

Property Value

A system-defined color with the value R:230 G:230 B:250 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LavenderBlush Property

Gets a system-defined color with the value R:255 G:240 B:245 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LavenderBlush { get; }
```

Property Value

A system-defined color with the value R:255 G:240 B:245 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LawnGreen Property

Gets a system-defined color with the value R:124 G:252 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LawnGreen { get; }
```

Property Value

A system-defined color with the value R:124 G:252 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LemonChiffon Property

Gets a system-defined color with the value R:255 G:250 B:205 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LemonChiffon { get; }
```

Property Value

A system-defined color with the value R:255 G:250 B:205 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightBlue Property

Gets a system-defined color with the value R:173 G:216 B:230 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightBlue { get; }
```

Property Value

A system-defined color with the value R:173 G:216 B:230 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightCoral Property

Gets a system-defined color with the value R:240 G:128 B:128 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightCoral { get; }
```

Property Value

A system-defined color with the value R:240 G:128 B:128 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightCyan Property

Gets a system-defined color with the value R:224 G:255 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightCyan { get; }
```

Property Value

A system-defined color with the value R:224 G:255 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightGoldenrodYellow Property

Gets a system-defined color with the value R:250 G:250 B:210 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightGoldenrodYellow { get; }
```

Property Value

A system-defined color with the value R:250 G:250 B:210 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightGray Property

Gets a system-defined color with the value R:211 G:211 B:211 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightGray { get; }
```

Property Value

A system-defined color with the value R:211 G:211 B:211 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightGreen Property

Gets a system-defined color with the value R:144 G:238 B:144 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightGreen { get; }
```

Property Value

A system-defined color with the value R:144 G:238 B:144 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightPink Property

Gets a system-defined color with the value R:255 G:182 B:193 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightPink { get; }
```

Property Value

A system-defined color with the value R:255 G:182 B:193 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightSalmon Property

Gets a system-defined color with the value R:255 G:160 B:122 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightSalmon { get; }
```

Property Value

A system-defined color with the value R:255 G:160 B:122 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightSeaGreen Property

Gets a system-defined color with the value R:32 G:178 B:170 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightSeaGreen { get; }
```

Property Value

A system-defined color with the value R:32 G:178 B:170 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightSkyBlue Property

Gets a system-defined color with the value R:135 G:206 B:250 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightSkyBlue { get; }
```

Property Value

A system-defined color with the value R:135 G:206 B:250 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightSlateGray Property

Gets a system-defined color with the value R:119 G:136 B:153 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightSlateGray { get; }
```

Property Value

A system-defined color with the value R:119 G:136 B:153 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightSteelBlue Property

Gets a system-defined color with the value R:176 G:196 B:222 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightSteelBlue { get; }
```

Property Value

A system-defined color with the value R:176 G:196 B:222 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LightYellow Property

Gets a system-defined color with the value R:255 G:255 B:224 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LightYellow { get; }
```

Property Value

A system-defined color with the value R:255 G:255 B:224 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Lime Property

Gets a system-defined color with the value R:0 G:255 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Lime { get; }
```

Property Value

A system-defined color with the value R:0 G:255 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.LimeGreen Property

Gets a system-defined color with the value R:50 G:205 B:50 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color LimeGreen { get; }
```

Property Value

A system-defined color with the value R:50 G:205 B:50 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Linen Property

Gets a system-defined color with the value R:250 G:240 B:230 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Linen { get; }
```

Property Value

A system-defined color with the value R:250 G:240 B:230 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Magenta Property

Gets a system-defined color with the value R:255 G:0 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Magenta { get; }
```

Property Value

A system-defined color with the value R:255 G:0 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Maroon Property

Gets a system-defined color with the value R:128 G:0 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Maroon { get; }
```

Property Value

A system-defined color with the value R:128 G:0 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumAquaMarine Property

Gets a system-defined color with the value R:102 G:205 B:170 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumAquaMarine { get; }
```

Property Value

A system-defined color with the value R:102 G:205 B:170 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumBlue Property

Gets a system-defined color with the value R:0 G:0 B:205 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumBlue { get; }
```

Property Value

A system-defined color with the value R:0 G:0 B:205 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumOrchid Property

Gets a system-defined color with the value R:186 G:85 B:211 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumOrchid { get; }
```

Property Value

A system-defined color with the value R:186 G:85 B:211 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumPurple Property

Gets a system-defined color with the value R:147 G:112 B:219 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumPurple { get; }
```

Property Value

A system-defined color with the value R:147 G:112 B:219 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumSeaGreen Property

Gets a system-defined color with the value R:60 G:179 B:113 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumSeaGreen { get; }
```

Property Value

A system-defined color with the value R:60 G:179 B:113 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumSlateBlue Property

Gets a system-defined color with the value R:123 G:104 B:238 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumSlateBlue { get; }
```

Property Value

A system-defined color with the value R:123 G:104 B:238 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumSpringGreen Property

Gets a system-defined color with the value R:0 G:250 B:154 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumSpringGreen { get; }
```

Property Value

A system-defined color with the value R:0 G:250 B:154 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumTurquoise Property

Gets a system-defined color with the value R:72 G:209 B:204 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumTurquoise { get; }
```

Property Value

A system-defined color with the value R:72 G:209 B:204 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MediumVioletRed Property

Gets a system-defined color with the value R:199 G:21 B:133 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MediumVioletRed { get; }
```

Property Value

A system-defined color with the value R:199 G:21 B:133 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MidnightBlue Property

Gets a system-defined color with the value R:25 G:25 B:112 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MidnightBlue { get; }
```

Property Value

A system-defined color with the value R:25 G:25 B:112 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MintCream Property

Gets a system-defined color with the value R:245 G:255 B:250 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MintCream { get; }
```

Property Value

A system-defined color with the value R:245 G:255 B:250 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.MistyRose Property

Gets a system-defined color with the value R:255 G:228 B:225 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color MistyRose { get; }
```

Property Value

A system-defined color with the value R:255 G:228 B:225 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Moccasin Property

Gets a system-defined color with the value R:255 G:228 B:181 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Moccasin { get; }
```

Property Value

A system-defined color with the value R:255 G:228 B:181 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.NavajoWhite Property

Gets a system-defined color with the value R:255 G:222 B:173 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color NavajoWhite { get; }
```

Property Value

A system-defined color with the value R:255 G:222 B:173 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Navy Property

Gets a system-defined color R:0 G:0 B:128 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Navy { get; }
```

Property Value

A system-defined color R:0 G:0 B:128 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.OldLace Property

Gets a system-defined color with the value R:253 G:245 B:230 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color OldLace { get; }
```

Property Value

A system-defined color with the value R:253 G:245 B:230 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Olive Property

Gets a system-defined color with the value R:128 G:128 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Olive { get; }
```

Property Value

A system-defined color with the value R:128 G:128 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.OliveDrab Property

Gets a system-defined color with the value R:107 G:142 B:35 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color OliveDrab { get; }
```

Property Value

A system-defined color with the value R:107 G:142 B:35 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Orange Property

Gets a system-defined color with the value R:255 G:165 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Orange { get; }
```

Property Value

A system-defined color with the value R:255 G:165 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.OrangeRed Property

Gets a system-defined color with the value R:255 G:69 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color OrangeRed { get; }
```

Property Value

A system-defined color with the value R:255 G:69 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Orchid Property

Gets a system-defined color with the value R:218 G:112 B:214 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Orchid { get; }
```

Property Value

A system-defined color with the value R:218 G:112 B:214 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PackedValue Property

Gets the current color as a packed value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The current color.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PaleGoldenrod Property

Gets a system-defined color with the value R:238 G:232 B:170 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PaleGoldenrod { get; }
```

Property Value

A system-defined color with the value R:238 G:232 B:170 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PaleGreen Property

Gets a system-defined color with the value R:152 G:251 B:152 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PaleGreen { get; }
```

Property Value

A system-defined color with the value R:152 G:251 B:152 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PaleTurquoise Property

Gets a system-defined color with the value R:175 G:238 B:238 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PaleTurquoise { get; }
```

Property Value

A system-defined color with the value R:175 G:238 B:238 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PaleVioletRed Property

Gets a system-defined color with the value R:219 G:112 B:147 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PaleVioletRed { get; }
```

Property Value

A system-defined color with the value R:219 G:112 B:147 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PapayaWhip Property

Gets a system-defined color with the value R:255 G:239 B:213 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PapayaWhip { get; }
```

Property Value

A system-defined color with the value R:255 G:239 B:213 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PeachPuff Property

Gets a system-defined color with the value R:255 G:218 B:185 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PeachPuff { get; }
```

Property Value

A system-defined color with the value R:255 G:218 B:185 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Peru Property

Gets a system-defined color with the value R:205 G:133 B:63 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Peru { get; }
```

Property Value

A system-defined color with the value R:205 G:133 B:63 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Pink Property

Gets a system-defined color with the value R:255 G:192 B:203 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Pink { get; }
```

Property Value

A system-defined color with the value R:255 G:192 B:203 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Plum Property

Gets a system-defined color with the value R:221 G:160 B:221 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Plum { get; }
```

Property Value

A system-defined color with the value R:221 G:160 B:221 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.PowderBlue Property

Gets a system-defined color with the value R:176 G:224 B:230 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color PowderBlue { get; }
```

Property Value

A system-defined color with the value R:176 G:224 B:230 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Purple Property

Gets a system-defined color with the value R:128 G:0 B:128 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Purple { get; }
```

Property Value

A system-defined color with the value R:128 G:0 B:128 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.R Property

Gets or sets the red component value of this [Color](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte R { get; set; }
```

Property Value

The red component value of this [Color](#).

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Red Property

Gets a system-defined color with the value R:255 G:0 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Red { get; }
```

Property Value

A system-defined color with the value R:255 G:0 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.RosyBrown Property

Gets a system-defined color with the value R:188 G:143 B:143 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color RosyBrown { get; }
```

Property Value

A system-defined color with the value R:188 G:143 B:143 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.RoyalBlue Property

Gets a system-defined color with the value R:65 G:105 B:225 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color RoyalBlue { get; }
```

Property Value

A system-defined color with the value R:65 G:105 B:225 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SaddleBrown Property

Gets a system-defined color with the value R:139 G:69 B:19 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SaddleBrown { get; }
```

Property Value

A system-defined color with the value R:139 G:69 B:19 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Salmon Property

Gets a system-defined color with the value R:250 G:128 B:114 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Salmon { get; }
```

Property Value

A system-defined color with the value R:250 G:128 B:114 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SandyBrown Property

Gets a system-defined color with the value R:244 G:164 B:96 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SandyBrown { get; }
```

Property Value

A system-defined color with the value R:244 G:164 B:96 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SeaGreen Property

Gets a system-defined color with the value R:46 G:139 B:87 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SeaGreen { get; }
```

Property Value

A system-defined color with the value R:46 G:139 B:87 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SeaShell Property

Gets a system-defined color with the value R:255 G:245 B:238 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SeaShell { get; }
```

Property Value

A system-defined color with the value R:255 G:245 B:238 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Sienna Property

Gets a system-defined color with the value R:160 G:82 B:45 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Sienna { get; }
```

Property Value

A system-defined color with the value R:160 G:82 B:45 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Silver Property

Gets a system-defined color with the value R:192 G:192 B:192 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Silver { get; }
```

Property Value

A system-defined color with the value R:192 G:192 B:192 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SkyBlue Property

Gets a system-defined color with the value R:135 G:206 B:235 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SkyBlue { get; }
```

Property Value

A system-defined color with the value R:135 G:206 B:235 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SlateBlue Property

Gets a system-defined color with the value R:106 G:90 B:205 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SlateBlue { get; }
```

Property Value

A system-defined color with the value R:106 G:90 B:205 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SlateGray Property

Gets a system-defined color with the value R:112 G:128 B:144 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SlateGray { get; }
```

Property Value

A system-defined color with the value R:112 G:128 B:144 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Snow Property

Gets a system-defined color with the value R:255 G:250 B:250 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Snow { get; }
```

Property Value

A system-defined color with the value R:255 G:250 B:250 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SpringGreen Property

Gets a system-defined color with the value R:0 G:255 B:127 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SpringGreen { get; }
```

Property Value

A system-defined color with the value R:0 G:255 B:127 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.SteelBlue Property

Gets a system-defined color with the value R:70 G:130 B:180 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color SteelBlue { get; }
```

Property Value

A system-defined color with the value R:70 G:130 B:180 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Tan Property

Gets a system-defined color with the value R:210 G:180 B:140 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Tan { get; }
```

Property Value

A system-defined color with the value R:210 G:180 B:140 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Teal Property

Gets a system-defined color with the value R:0 G:128 B:128 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Teal { get; }
```

Property Value

A system-defined color with the value R:0 G:128 B:128 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Thistle Property

Gets a system-defined color with the value R:216 G:191 B:216 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Thistle { get; }
```

Property Value

A system-defined color with the value R:216 G:191 B:216 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Tomato Property

Gets a system-defined color with the value R:255 G:99 B:71 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Tomato { get; }
```

Property Value

A system-defined color with the value R:255 G:99 B:71 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.TransparentBlack Property

Gets a system-defined color with the value R:0 G:0 B:0 A:0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color TransparentBlack { get; }
```

Property Value

A system-defined color with the value R:0 G:0 B:0 A:0.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.TransparentWhite Property

Gets a system-defined color with the value R:255 G:255 B:255 A:0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color TransparentWhite { get; }
```

Property Value

A system-defined color with the value R:255 G:255 B:255 A:0.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Turquoise Property

Gets a system-defined color with the value R:64 G:224 B:208 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Turquoise { get; }
```

Property Value

A system-defined color with the value R:64 G:224 B:208 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Violet Property

Gets a system-defined color with the value R:238 G:130 B:238 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Violet { get; }
```

Property Value

A system-defined color with the value R:238 G:130 B:238 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Wheat Property

Gets a system-defined color with the value R:245 G:222 B:179 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Wheat { get; }
```

Property Value

A system-defined color with the value R:245 G:222 B:179 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.White Property

Gets a system-defined color with the value R:255 G:255 B:255 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color White { get; }
```

Property Value

A system-defined color with the value R:255 G:255 B:255 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.WhiteSmoke Property

Gets a system-defined color with the value R:245 G:245 B:245 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color WhiteSmoke { get; }
```

Property Value

A system-defined color with the value R:245 G:245 B:245 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.Yellow Property

Gets a system-defined color with the value R:255 G:255 B:0 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color Yellow { get; }
```

Property Value

A system-defined color with the value R:255 G:255 B:0 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Color.YellowGreen Property

Gets a system-defined color with the value R:154 G:205 B:50 A:255.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Color YellowGreen { get; }
```

Property Value

A system-defined color with the value R:154 G:205 B:50 A:255.

See Also

Reference

[Color Structure](#)

[Color Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ColorWriteChannels Enumeration

Defines the color channels that can be chosen for a per-channel write to a render target color buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum ColorWriteChannels
```

Members

Member name	Description
All	All buffer channels.
Alpha	Alpha channel of a buffer.
Blue	Blue channel of a buffer.
Green	Green channel of a buffer.
None	No channel selected.
Red	Red channel of a buffer.

See Also

Reference

[RenderState.ColorWriteChannels Property](#)

[RenderState.ColorWriteChannels1 Property](#)

[RenderState.ColorWriteChannels2 Property](#)

[RenderState.ColorWriteChannels3 Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

CompareFunction Enumeration

Defines comparison functions that can be chosen for alpha, stencil, or depth-buffer tests.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CompareFunction
```

Members

Member name	Description
Always	Always pass the test.
Equal	Accept the new pixel if its value is equal to the value of the current pixel.
Greater	Accept the new pixel if its value is greater than the value of the current pixel.
GreaterEqual	Accept the new pixel if its value is greater than or equal to the value of the current pixel.
Less	Accept the new pixel if its value is less than the value of the current pixel.
LessEqual	Accept the new pixel if its value is less than or equal to the value of the current pixel.
Never	Always fail the test.
NotEqual	Accept the new pixel if its value does not equal the value of the current pixel.

See Also

Reference

[AlphaFunction](#)

[CounterClockwiseStencilFunction](#)

[DepthBufferFunction](#)

[StencilFunction](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

CompilationFailedException Class

Note

This class is available only when developing for Windows.

The exception that is thrown if the compilation of an effect fails.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class CompilationFailedException : ExternalException
```

See Also

Reference

[CompilationFailedException Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista









CompilationFailedException Members

The following tables list the members exposed by the `CompilationFailedException` type.


Public Constructors

Name	Description
 CompilationFailedException	Overloaded. Initializes a new instance of the CompilationFailedException class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompilationFailedException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilationFailedException Constructor

Initializes a new instance of the [CompilationFailedException](#) class.

Overload List

Name	Description
CompilationFailedException ()	Initializes a new instance of this class.
CompilationFailedException (String)	Initializes a new instance of this class with a specified error message.
CompilationFailedException (String, Exception)	Initializes a new instance of the CompilationFailedException class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[CompilationFailedException Class](#)

[CompilationFailedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilationFailedException Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompilationFailedException ()
```

See Also

Reference

[CompilationFailedException Class](#)

[CompilationFailedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilationFailedException Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompilationFailedException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [CompilationFailedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[CompilationFailedException Class](#)

[CompilationFailedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

CompilationFailedException Constructor (String, Exception)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [CompilationFailedException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompilationFailedException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [CompilationFailedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[CompilationFailedException Class](#)







[CompilationFailedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

CompilationFailedException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[CompilationFailedException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilationFailedException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[CompilationFailedException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledEffect Structure

Note

This structure is available only when developing for Windows.

Represents a compiled [Effect](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct CompiledEffect
```

See Also

Reference

[CompiledEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



CompiledEffect Members

The following tables list the members exposed by the CompiledEffect type.






Public Constructors

Name	Description
 CompiledEffect	Creates a new instance of the CompiledEffect class.



Public Properties

Name	Description
 ErrorsAndWarnings	Gets any errors and warnings generated by compiling the effect.
 Success	Indicates whether the effect was compiled successfully.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetEffectCode	Gets the compiled byte code for this shader.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompiledEffect Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledEffect Constructor

Note

This constructor is available only when developing for Windows.

Creates a new instance of the [CompiledEffect](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompiledEffect (  
    byte[] compiledEffectCode,  
    string errors  
)
```

Parameters

compiledEffectCode

The compiled effect code.

errors

A string to receive compilation errors, if any.

See Also

Reference

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[CompiledEffect Structure](#)






[CompiledEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

CompiledEffect Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEffectCode	Gets the compiled byte code for this shader.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompiledEffect Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledEffect.GetEffectCode Method

Note

This method is available only when developing for Windows.

Gets the compiled byte code for this shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte[] GetEffectCode ()
```

Return Value

The compiled bytecode.

See Also

Reference

[CompiledEffect Structure](#)

[CompiledEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledEffect.ToString Method

Note

This method is available only when developing for Windows.

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[CompiledEffect Structure](#)



[CompiledEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledEffect Properties

Public Properties

	Name	Description
	ErrorsAndWarnings	Gets any errors and warnings generated by compiling the effect.
	Success	Indicates whether the effect was compiled successfully.

See Also

Reference

[CompiledEffect Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledEffect.ErrorsAndWarnings Property

Note

This property is available only when developing for Windows.

Gets any errors and warnings generated by compiling the effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string ErrorsAndWarnings { get; }
```

Property Value

A string containing errors and warnings from compiling the effect.

See Also

Reference

[CompiledEffect Structure](#)

[CompiledEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledEffect.Success Property

Note

This property is available only when developing for Windows.

Indicates whether the effect was compiled successfully.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Success { get; }
```

Property Value

true if the effect was compiled successfully; **false** otherwise.

See Also

Reference

[CompiledEffect Structure](#)

[CompiledEffect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader Structure

Note

This structure is available only when developing for Windows.

Represents a compiled shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct CompiledShader
```

See Also

Reference

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista





CompiledShader Members

The following tables list the members exposed by the CompiledShader type.









Public Constructors

Name	Description
 CompiledShader	Initializes a new instance of the CompiledShader class.



Public Properties

Name	Description
 ErrorsAndWarnings	Gets the first compile error message that occurred.
 ShaderSize	Gets the size of the shader byte code, in bytes.
 ShaderVersion	Gets the shader version of the compiled shader.
 Success	Returns a value indicating whether the shader compilation was successful.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetInputSemantics	Gets the semantics for the shader inputs. Use this method to determine the input vertex format.
 GetOutputSemantics	Gets the semantics for all shader output elements.
 GetSamplers	Gets the sampler names referenced in a shader.
 GetShaderCode	Gets the compiled shader code.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a String that represents the current CompiledShader .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompiledShader Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledShader Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [CompiledShader](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompiledShader (  
    byte[] compiledShaderCode,  
    string errors  
)
```

Parameters

compiledShaderCode

The compiled shader to encapsulate.

errors

String containing at least the first compile error message that occurred. This includes effect compiler errors and high-level language compile errors.

See Also

Reference

[CompiledShader Structure](#)









[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

CompiledShader Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetInputSemantics	Gets the semantics for the shader inputs. Use this method to determine the input vertex format.
	GetOutputSemantics	Gets the semantics for all shader output elements.
	GetSamplers	Gets the sampler names referenced in a shader.
	GetShaderCode	Gets the compiled shader code.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current CompiledShader .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompiledShader Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledShader.GetInputSemantics Method

Note

This method is available only when developing for Windows.

Gets the semantics for the shader inputs. Use this method to determine the input vertex format.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderSemantic[] GetInputSemantics ()
```

Return Value

An array of shader semantics.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.GetOutputSemantics Method

Note

This method is available only when developing for Windows.

Gets the semantics for all shader output elements.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderSemantic[] GetOutputSemantics ()
```

Return Value

An array of shader semantics.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.GetSamplers Method

Note

This method is available only when developing for Windows.

Gets the sampler names referenced in a shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string[] GetSamplers ()
```

Return Value

An array of sampler names.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.GetShaderCode Method

Note

This method is available only when developing for Windows.

Gets the compiled shader code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte[] GetShaderCode ()
```

Return Value

The compiled shader code.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.ToString Method

Note

This method is available only when developing for Windows.

Returns a [String](#) that represents the current [CompiledShader](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

A [String](#) that represents the current [CompiledShader](#).

See Also

Reference

[CompiledShader Structure](#)





[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader Properties

Public Properties

	Name	Description
	ErrorsAndWarnings	Gets the first compile error message that occurred.
	ShaderSize	Gets the size of the shader byte code, in bytes.
	ShaderVersion	Gets the shader version of the compiled shader.
	Success	Returns a value indicating whether the shader compilation was successful.

See Also

Reference

[CompiledShader Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompiledShader.ErrorsAndWarnings Property

Note

This property is available only when developing for Windows.

Gets the first compile error message that occurred.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string ErrorsAndWarnings { get; }
```

Property Value

The first compile error message that occurred. This includes effect compiler errors and high-level language compile errors.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.ShaderSize Property

Note

This property is available only when developing for Windows.

Gets the size of the shader byte code, in bytes.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ShaderSize { get; }
```

Property Value

The size of the shader byte code, in bytes.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.ShaderVersion Property

Note

This property is available only when developing for Windows.

Gets the shader version of the compiled shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Version ShaderVersion { get; }
```

Property Value

The shader version of the compiled shader.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompiledShader.Success Property

Note

This property is available only when developing for Windows.

Returns a value indicating whether the shader compilation was successful.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Success { get; }
```

Property Value

true if the shader compilation succeeds; **false** otherwise.

See Also

Reference

[CompiledShader Structure](#)

[CompiledShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerIncludeHandler Class

Note

This class is available only when developing for Windows.

The abstract base class for custom compiler include file handlers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class CompilerIncludeHandler : IDisposable
```

See Also

Reference

[CompilerIncludeHandler Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista








CompilerIncludeHandler Members

The following tables list the members exposed by the CompilerIncludeHandler type.



Public Constructors

Name	Description
 CompilerIncludeHandler	Initializes a new instance of this class.

Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Open	Returns a Stream containing include data from a specified file.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompilerIncludeHandler Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilerIncludeHandler Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompilerIncludeHandler ()
```

See Also

Reference

[CompilerIncludeHandler Class](#)








[CompilerIncludeHandler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

CompilerIncludeHandler Methods

Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Open	Returns a Stream containing include data from a specified file.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompilerIncludeHandler Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilerIncludeHandler.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
CompilerIncludeHandler.Dispose ()	Immediately releases the unmanaged resources used by this object.
CompilerIncludeHandler.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[CompilerIncludeHandler Class](#)

[CompilerIncludeHandler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilerIncludeHandler.Dispose Method ()

Note

This method is available only when developing for Windows.

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[CompilerIncludeHandler Class](#)

[CompilerIncludeHandler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerIncludeHandler.Dispose Method (Boolean)

Note

This method is available only when developing for Windows.

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) This parameter marshals as an unsigned byte.

See Also

Reference

[CompilerIncludeHandler Class](#)

[CompilerIncludeHandler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerIncludeHandler.Finalize Method

Note

This method is available only when developing for Windows.

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[CompilerIncludeHandler Class](#)

[CompilerIncludeHandler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerIncludeHandler.Open Method

Note

This method is available only when developing for Windows.

Returns a [Stream](#) containing include data from a specified file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract Stream Open (  
    CompilerIncludeHandlerType includeType,  
    string filename  
)
```

Parameters

includeType

A [CompilerIncludeHandlerType](#) indicating whether the file is a local or system resource.

filename

The file name of the file to open.

Return Value

A [Stream](#) of compiler include data.

See Also

Reference

[CompilerIncludeHandler Class](#)

[CompilerIncludeHandler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerIncludeHandlerType Enumeration

Note

This enumeration is available only when developing for Windows.

Identifies an include file as a local or system resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CompilerIncludeHandlerType
```

Members

Member name	Description
Local	A local resource.
System	A system resource.

See Also

Reference

[CompilerIncludeHandler.Open Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerMacro Structure

Note

This structure is available only when developing for Windows.

Represents a compiler macro.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct CompilerMacro
```

See Also

Reference

[CompilerMacro Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista





CompilerMacro Members

The following tables list the members exposed by the CompilerMacro type.



Public Properties

	Name	Description
	Definition	Gets or sets the macro definition.
	Name	Gets or sets the macro name.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[CompilerMacro Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilerMacro Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CompilerMacro Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilerMacro.ToString Method

Note

This method is available only when developing for Windows.

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[CompilerMacro Structure](#)



[CompilerMacro Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerMacro Properties

Public Properties

	Name	Description
	Definition	Gets or sets the macro definition.
	Name	Gets or sets the macro name.

See Also

Reference

[CompilerMacro Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

CompilerMacro.Definition Property

Note

This property is available only when developing for Windows.

Gets or sets the macro definition.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Definition { get; set; }
```

Property Value

The definition of the macro.

See Also

Reference

[CompilerMacro Structure](#)

[CompilerMacro Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerMacro.Name Property

Note

This property is available only when developing for Windows.

Gets or sets the macro name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of the macro.

See Also

Reference

[CompilerMacro Structure](#)

[CompilerMacro Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CompilerOptions Enumeration

Defines optimization options that may be chosen for shader and effect code compilation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum CompilerOptions
```

Members

Member name	Description
AvoidFlowControl	Hints to the compiler to avoid using flow-control instructions.
Debug	Inserts debug file name, line numbers, and type and symbol information during shader compile.
ForcePixelShaderSoftwareNoOptimizations	Forces the compiler to compile against the next highest available software target for pixel shaders. This flag also turns optimizations off and debugging on.
ForceVertexShaderSoftwareNoOptimizations	Forces the compiler to compile against the next highest available software target for vertex shaders. This flag also turns optimizations off and debugging on.
None	No options specified.
NoPreShader	Disables preshaders. The compiler will not pull out static expressions for evaluation on the host CPU. Additionally, the compiler will not lift any expressions when compiling stand-alone functions.
NotCloneable	Indicates the effect will be non-cloneable and will not contain any shader binary data. Setting this flag reduces effect memory usage by about 50 percent because it eliminates the need for the effect system to keep a copy of the shaders in memory.
PackMatrixColumnMajor	Unless explicitly specified, matrices will be packed in column major order (each vector will be in a single column) when passed to and from the shader. This is generally more efficient because it allows vector-matrix multiplication to be performed using a series of dot products.
PackMatrixRowMajor	Unless explicitly specified, matrices will be packed in row major order (each vector will be in a single row) when passed to or from the shader.
PartialPrecision	Forces all computations in the resulting shader to occur at partial precision. This may result in faster evaluation of shaders on some hardware.
PreferFlowControl	Hints to the compiler to prefer using flow-control instructions.
SkipOptimization	Instructs the compiler to skip optimization steps during code generation. Unless you are trying to isolate a problem in your code and you suspect the compiler, using this option is not recommended.
SkipValidation	Do not validate the generated code against known capabilities and constraints. This option is recommended only when compiling shaders that are known to work (that is, shaders that have compiled before without this option). Shaders are always validated by the runtime before they are set to the device.

See Also

Reference

[Effect Constructor](#)

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[ShaderCompiler.AssembleFromFile Method](#)

[ShaderCompiler.AssembleFromSource Method](#)

[ShaderCompiler.CompileFromFile Method](#)

[ShaderCompiler.CompileFromSource Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

CubeMapFace Enumeration

Defines the faces of a cube map in the [TextureCube](#) class type.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CubeMapFace
```

Members

Member name	Description
NegativeX	Negative x-face of the cube map.
NegativeY	Negative y-face of the cube map.
NegativeZ	Negative z-face of the cube map.
PositiveX	Positive x-face of the cube map.
PositiveY	Positive y-face of the cube map.
PositiveZ	Positive z-face of the cube map.

See Also

Reference

[TextureCube.GetData Method](#)

[TextureCube.SetData Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

CullMode Enumeration

Defines winding orders that may be used to identify back faces for culling.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum CullMode
```

Members

Member name	Description
CullClockwiseFace	Cull back faces with clockwise vertices.
CullCounterClockwiseFace	Cull back faces with counterclockwise vertices.
None	Do not cull back faces.

See Also

Reference

[RenderState.CullMode Property](#)

[RenderState.TwoSidedStencilMode Property](#)

[GraphicsDeviceCapabilities.PrimitiveCaps.SupportsCullClockwiseFace Property](#)

[GraphicsDeviceCapabilities.PrimitiveCaps.SupportsCullCounterClockwiseFace Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthFormat Enumeration

Defines the format of data in a depth buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum DepthFormat
```

Members

Member name	Description
Depth15Stencil1	A 16-bit depth-buffer bit depth in which 15 bits are reserved for the depth channel and 1 bit is reserved for the stencil channel.
Depth16	A 16-bit depth-buffer bit depth.
Depth24	A 32-bit depth-buffer bit depth that uses 24 bits for the depth channel.
Depth24Stencil4	A 32-bit depth-buffer bit depth that uses 24 bits for the depth channel and 4 bits for the stencil channel.
Depth24Stencil8	A non-lockable format that contains 24 bits of depth (in a 24-bit floating-point format – 20E4) and 8 bits of stencil.
Depth24Stencil8Single	A 32-bit depth-buffer bit depth that uses 24 bits for the depth channel and 8 bits for the stencil channel.
Depth32	a 32-bit depth-buffer bit depth.
Unknown	Format is unknown.

RemarksAll depth-stencil formats except **Depth16Lockable** indicate no particular bit ordering per pixel, and the driver is allowed to consume more than the indicated number of bits per depth channel (but not stencil channel).

See Also

Concepts

[What Is a Depth Buffer?](#)

[What Is a Stencil Buffer?](#)

Tasks

[How To: Create a Depth Texture](#)

Reference

[CheckDepthStencilMatch](#)

[CheckDeviceFormat](#)

[AutoDepthStencilFormat](#)

[PresentOptions](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

DepthStencilBuffer Class

Queries and prepares depth stencil buffers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class DepthStencilBuffer : IDisposable
```

Remarks

A depth stencil buffer is where depth and stencil data are kept for a render target. The depth buffer is used to determine which 3D objects appear behind other 3D objects; the stencil buffer is used for special effects. Depth stencil buffers are often created at the same time render targets are created. Depth stencil buffers must be recreated when the device is reset. The [DepthStencilBuffer](#) property on [GraphicsDevice](#) points to the current **DepthStencilBuffer**.

To use a **DepthStencilBuffer**, you must:

1. Create the **DepthStencilBuffer**.

C#

```
shadowDepthBuffer =
    GfxComponent.CreateDepthStencil(shadowRenderTarget,
    DepthFormat.Depth24Stencil8Single);
```

C#

```
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target)
{
    return new DepthStencilBuffer(target.GraphicsDevice, target.Width,
        target.Height, target.GraphicsDevice.DepthStencilBuffer.Format,
        target.MultiSampleType, target.MultiSampleQuality);
}
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target, DepthFormat depth)
{
    if (GraphicsAdapter.DefaultAdapter.CheckDepthStencilMatch(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        target.Format,
        depth))
    {
        return new DepthStencilBuffer(target.GraphicsDevice,
            target.Width, target.Height, depth,
            target.MultiSampleType, target.MultiSampleQuality);
    }
    else
        return CreateDepthStencil(target);
}
```

2. Set the **DepthStencilBuffer** using [DepthStencilBuffer](#). It is a good idea to cache the current **DepthStencilBuffer** beforehand.

C#

```
// Cache the current depth buffer
DepthStencilBuffer old = GraphicsDevice.DepthStencilBuffer;
// Set our custom depth buffer
```

```
GraphicsDevice.DepthStencilBuffer = shadowDepthBuffer;
```

3. Draw into your **DepthStencilBuffer**.

C#

```
// Render the shadow map  
GraphicsDevice.Clear(Color.Black);  
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **DepthStencilBuffer**.

C#

```
// Reset the depth buffer  
GraphicsDevice.DepthStencilBuffer = old;
```

Custom depth stencil buffers are normally used in conjunction with a custom [RenderTarget](#). [How To: Create a Depth Texture](#) has an example of this.

See Also

Concepts

[What Is a Depth Buffer?](#)

[What Is a Stencil Buffer?](#)

Tasks

[How To: Load Content](#)

[How To: Draw a Shadow](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[DepthStencilBuffer](#)

[RenderTarget](#)

[GraphicsDeviceManager.DeviceReset Event](#)

[DepthStencilBuffer Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista











DepthStencilBuffer Members

The following tables list the members exposed by the DepthStencilBuffer type.







Public Constructors

Name	Description
 DepthStencilBuffer	Overloaded. Initializes a new instance of the DepthStencilBuffer class.





Public Properties

Name	Description
 Format	Gets the data format of this depth stencil buffer.
 GraphicsDevice	Gets the graphics device associated with this depth stencil buffer.
 Height	Gets the height, in pixels, of the surface.
 IsContentLost	Determines if the buffer data has been lost due to a lost device event.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 MultiSampleQuality	Gets the number of quality stops available for a given multisample type.
 MultiSampleType	Gets the levels of full-scene multisampling that the device can apply.
 Name	Gets the name of this depth stencil buffer.
 Tag	Gets the resource tag for this depth stencil buffer.
 Width	Returns the width, in pixels, of this DepthStencilBuffer .



Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs when content is about to be lost on a GraphicsDevice .
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 ContentLost	Occurs when resources are lost (for example, when the current device is lost).
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[DepthStencilBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DepthStencilBuffer Constructor

Initializes a new instance of the [DepthStencilBuffer](#) class.

Overload List

Name	Description
DepthStencilBuffer (GraphicsDevice, Int32, Int32, DepthFormat)	Initializes a new instance of the DepthStencilBuffer class.
DepthStencilBuffer (GraphicsDevice, Int32, Int32, DepthFormat, MultiSampleType, Int32)	Initializes a new instance of the DepthStencilBuffer class, specifying the multisample type and quality.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DepthStencilBuffer Constructor (GraphicsDevice, Int32, Int32, DepthFormat)

Initializes a new instance of the [DepthStencilBuffer](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DepthStencilBuffer (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    DepthFormat format
)
```

Parameters

graphicsDevice

The graphics device that will display the depth stencil buffer.

width

Width, in pixels, of the depth stencil buffer. You can use **graphicsDevice.PresentationParameters.BackBufferWidth** to get the current screen width.

height

Height, in pixels, of the depth stencil buffer. You can use **graphicsDevice.PresentationParameters.BackBufferHeight** to get the current screen height.

format

The format of the data to be placed in this depth stencil buffer. Use [CheckDepthStencilMatch](#) to determine if a particular depth format is usable in a render target.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>width</i> and <i>height</i> are less than or equal to zero. <i>width</i> and <i>height</i> must be greater than zero.
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> The device does not support creating a depth buffer of the given format. <i>format</i> is DepthFormat.Unknown. Textures cannot be created using DepthFormat.Unknown. <i>width</i> is larger than MaxTextureWidth. <i>height</i> is larger than MaxTextureHeight.
InvalidOperationException	Unable to create this DepthStencilBuffer resource.

Example

C#

```
shadowDepthBuffer =
    GfxComponent.CreateDepthStencil(shadowRenderTarget,
    DepthFormat.Depth24Stencil8Single);
```

C#

```
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target)
{
    return new DepthStencilBuffer(target.GraphicsDevice, target.Width,
```

```
        target.Height, target.GraphicsDevice.DepthStencilBuffer.Format,
        target.MultiSampleType, target.MultiSampleQuality);
    }
    public static DepthStencilBuffer CreateDepthStencil(
        RenderTarget2D target, DepthFormat depth)
    {
        if (GraphicsAdapter.DefaultAdapter.CheckDepthStencilMatch(
            DeviceType.Hardware,
            GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
            target.Format,
            depth))
        {
            return new DepthStencilBuffer(target.GraphicsDevice,
                target.Width, target.Height, depth,
                target.MultiSampleType, target.MultiSampleQuality);
        }
        else
            return CreateDepthStencil(target);
    }
}
```

See Also

Tasks

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[CheckDepthStencilMatch](#)

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer Constructor (GraphicsDevice, Int32, Int32, DepthFormat, MultiSampleType, Int32)

Initializes a new instance of the [DepthStencilBuffer](#) class, specifying the multisample type and quality.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DepthStencilBuffer (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    DepthFormat format,
    MultiSampleType multiSampleType,
    int multiSampleQuality
)
```

Parameters

graphicsDevice

The graphics device that will display the depth stencil buffer.

width

Width, in pixels, of the depth stencil buffer. You can use **graphicsDevice.PresentationParameters.BackBufferWidth** to get the current screen width.

height

Height, in pixels, of the depth stencil buffer. You can use **graphicsDevice.PresentationParameters.BackBufferHeight** to get the current screen height.

format

The format of the data to be placed in this depth stencil buffer. Use [CheckDepthStencilMatch](#) to determine if a particular depth format is usable in a render target.

multiSampleType

The levels of full-scene multisampling that the device can apply. You can use **graphicsDevice.PresentationParameters.MultiSampleType** to get the [MultiSampleType](#) of the back buffer.

multiSampleQuality

The number of quality stops available for a given multisample type. You can use **graphicsDevice.PresentationParameters.MultiSampleQuality** to get the multisample quality of the back buffer.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>width</i> and <i>height</i> are less than or equal to zero. <i>width</i> and <i>height</i> must be greater than zero.
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> The graphics device does not support creating a depth buffer of the given format. <i>format</i> is DepthFormat.Unknown. Textures cannot be created using DepthFormat.Unknown. <i>width</i> is larger than MaxTextureWidth. <i>height</i> is larger than MaxTextureHeight.
InvalidOperationException	Unable to create this DepthStencilBuffer resource.

Example

C#

```
shadowDepthBuffer =
    GfxComponent.CreateDepthStencil(shadowRenderTarget,
```

```
DepthFormat.Depth24Stencil8Single);
```

C#

```
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target)
{
    return new DepthStencilBuffer(target.GraphicsDevice, target.Width,
        target.Height, target.GraphicsDevice.DepthStencilBuffer.Format,
        target.MultiSampleType, target.MultiSampleQuality);
}
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target, DepthFormat depth)
{
    if (GraphicsAdapter.DefaultAdapter.CheckDepthStencilMatch(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        target.Format,
        depth))
    {
        return new DepthStencilBuffer(target.GraphicsDevice,
            target.Width, target.Height, depth,
            target.MultiSampleType, target.MultiSampleQuality);
    }
    else
        return CreateDepthStencil(target);
}
```

See Also

Tasks

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[CheckDepthStencilMatch](#)

[DepthStencilBuffer Class](#)







[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_ContentLost	Occurs when content is about to be lost on a GraphicsDevice .
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[DepthStencilBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DepthStencilBuffer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
DepthStencilBuffer.Dispose ()	Immediately releases the unmanaged resources used by this object.
DepthStencilBuffer.Dispose (Boolean)	Releases the unmanaged resources used by this object and optionally releases the managed resources.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DepthStencilBuffer.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Dispose Method (Boolean)

Releases the unmanaged resources used by this object and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [DepthStencilBuffer](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

RemarksThis method overrides [Finalize](#). Application code should not call this method; an object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.raise_ContentLost Method

Note

This method is available only when developing for Windows.

Occurs when content is about to be lost on a [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void raise_ContentLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

DepthStencilBuffer.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[DepthStencilBuffer Class](#)











[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

DepthStencilBuffer Properties

Public Properties

	Name	Description
	Format	Gets the data format of this depth stencil buffer.
	GraphicsDevice	Gets the graphics device associated with this depth stencil buffer.
	Height	Gets the height, in pixels, of the surface.
	IsContentLost	Determines if the buffer data has been lost due to a lost device event.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	MultiSampleQuality	Gets the number of quality stops available for a given multisample type.
	MultiSampleType	Gets the levels of full-scene multisampling that the device can apply.
	Name	Gets the name of this depth stencil buffer.
	Tag	Gets the resource tag for this depth stencil buffer.
	Width	Returns the width, in pixels, of this DepthStencilBuffer .

See Also

Reference

[DepthStencilBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DepthStencilBuffer.Format Property

Gets the data format of this depth stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DepthFormat Format { get; }
```

Property Value

The data format of this depth stencil buffer.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.GraphicsDevice Property

Gets the graphics device associated with this depth stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The graphics device associated with this depth stencil buffer.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Height Property

Gets the height, in pixels, of the surface.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

The height, in pixels, of the surface.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.IsContentLost Property

Determines if the buffer data has been lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsContentLost { get; }
```

Property Value

true if the content was lost; **false** otherwise.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed. **false** if the object is not disposed.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.MultiSampleQuality Property

Gets the number of quality stops available for a given multisample type.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MultiSampleQuality { get; }
```

Property Value

The number of quality stops available for a given multisample type.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.MultiSampleType Property

Gets the levels of full-scene multisampling that the device can apply.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MultiSampleType MultiSampleType { get; }
```

Property Value

The levels of full-scene multisampling that the device can apply.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Name Property

Gets the name of this depth stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of this depth stencil buffer.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Tag Property

Gets the resource tag for this depth stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tag for this depth stencil buffer.

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Width Property

Returns the width, in pixels, of this [DepthStencilBuffer](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

The width, in pixels, of this [DepthStencilBuffer](#).

See Also

Reference

[DepthStencilBuffer Class](#)



[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer Events

Public Events

	Name	Description
	Content Lost	Occurs when resources are lost (for example, when the current device is lost).
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[DepthStencilBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DepthStencilBuffer.ContentLost Event

Occurs when resources are lost (for example, when the current device is lost).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual event EventHandler ContentLost
```

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DepthStencilBuffer.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[DepthStencilBuffer Class](#)

[DepthStencilBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DeviceLostException Class

The exception that is thrown when the device has been lost, but cannot be reset at this time. Therefore, rendering is not possible.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class DeviceLostException : ExternalException
```

See Also

Reference

[DeviceLostException Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









DeviceLostException Members

The following tables list the members exposed by the DeviceLostException type.


Public Constructors

	Name	Description
	DeviceLostException	Overloaded. Initializes a new instance of this class.







Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)



Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DeviceLostException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceLostException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
DeviceLostException ()	Initializes a new instance of this class.
DeviceLostException (String)	Initializes a new instance of this class with a specified error message.
DeviceLostException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[DeviceLostException Class](#)

[DeviceLostException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceLostException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceLostException ()
```

Remarks

This constructor initializes the [Message](#) property of the new instance to a system-supplied message that describes the error, such as "The device has been lost but cannot be reset at this time." This message takes into account the current system culture.

The following table shows the initial property values for an instance of [DeviceLostException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceLostException Class](#)

[DeviceLostException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceLostException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceLostException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DeviceLostException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceLostException Class](#)

[DeviceLostException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceLostException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceLostException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [DeviceLostException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceLostException Class](#)







[DeviceLostException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceLostException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[DeviceLostException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceLostException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[DeviceLostException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotResetException Class

The exception that is thrown when the device has been lost, but can be reset at this time.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class DeviceNotResetException : ExternalException
```

See Also

Reference

[DeviceNotResetException Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









DeviceNotResetException Members

The following tables list the members exposed by the DeviceNotResetException type.


Public Constructors

Name	Description
 DeviceNotResetException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DeviceNotResetException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotResetException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
DeviceNotResetException ()	Initializes a new instance of this class.
DeviceNotResetException (String)	Initializes a new instance of this class with a specified error message.
DeviceNotResetException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[DeviceNotResetException Class](#)

[DeviceNotResetException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotResetException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceNotResetException ()
```

Remarks

This constructor initializes the [Message](#) property of the new instance to a system-supplied message that describes the error, such as "The device has been lost but can be reset at this time." This message takes into account the current system culture.

The following table shows the initial property values for an instance of [DeviceNotResetException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceNotResetException Class](#)

[DeviceNotResetException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceNotResetException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceNotResetException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DeviceNotResetException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceNotResetException Class](#)

[DeviceNotResetException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceNotResetException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceNotResetException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [DeviceNotResetException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceNotResetException Class](#)







[DeviceNotResetException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceNotResetException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[DeviceNotResetException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotResetException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[DeviceNotResetException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotSupportedException Class

The exception that is thrown when the graphics device does not support the requested capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class DeviceNotSupportedException : ExternalException
```

See Also

Reference

[DeviceNotSupportedException Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









DeviceNotSupportedException Members

The following tables list the members exposed by the DeviceNotSupportedException type.


Public Constructors

Name	Description
 DeviceNotSupportedException	Overloaded. Initializes a new instance of the DeviceNotSupportedException class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DeviceNotSupportedException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotSupportedException Constructor

Initializes a new instance of the [DeviceNotSupportedException](#) class.

Overload List

Name	Description
DeviceNotSupportedException ()	Initializes a new instance of the DeviceNotSupportedException class.
DeviceNotSupportedException (String)	Initializes a new instance of the DeviceNotSupportedException class with a specified error message.
DeviceNotSupportedException (String, Exception)	Initializes a new instance of the DeviceNotSupportedException class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[DeviceNotSupportedException Class](#)

[DeviceNotSupportedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotSupportedException Constructor ()

Initializes a new instance of the [DeviceNotSupportedException](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceNotSupportedException ()
```

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DeviceNotSupportedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceNotSupportedException Class](#)

[DeviceNotSupportedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceNotSupportedException Constructor (String)

Initializes a new instance of the [DeviceNotSupportedException](#) class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceNotSupportedException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DeviceNotSupportedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceNotSupportedException Class](#)

[DeviceNotSupportedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceNotSupportedException Constructor (String, Exception)

Initializes a new instance of the [DeviceNotSupportedException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceNotSupportedException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the inner parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DeviceNotSupportedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceNotSupportedException Class](#)







[DeviceNotSupportedException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceNotSupportedException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[DeviceNotSupportedException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceNotSupportedException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[DeviceNotSupportedException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceStillDrawingException Class

The error that is thrown when the device is still drawing.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class DeviceStillDrawingException : ExternalException
```

See Also

Reference

[DeviceStillDrawingException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









DeviceStillDrawingException Members

The following tables list the members exposed by the DeviceStillDrawingException type.


Public Constructors

Name	Description
 DeviceStillDrawingException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DeviceStillDrawingException Class](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceStillDrawingException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
DeviceStillDrawingException ()	Initializes a new instance of this class.
DeviceStillDrawingException (String)	Initializes a new instance of this class with a specified error message.
DeviceStillDrawingException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[DeviceStillDrawingException Class](#)

[DeviceStillDrawingException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceStillDrawingException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceStillDrawingException ()
```

Remarks

This constructor initializes the [Message](#) property of the new instance to a system-supplied message that describes the error, such as "The device has been lost but cannot be reset at this time." This message takes into account the current system culture.

The following table shows the initial property values for an instance of [DeviceStillDrawingException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceStillDrawingException Class](#)

[DeviceStillDrawingException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceStillDrawingException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceStillDrawingException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DeviceStillDrawingException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceStillDrawingException Class](#)

[DeviceStillDrawingException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceStillDrawingException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceStillDrawingException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [DeviceStillDrawingException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DeviceStillDrawingException Class](#)







[DeviceStillDrawingException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DeviceStillDrawingException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[DeviceStillDrawingException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceStillDrawingException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[DeviceStillDrawingException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DeviceType Enumeration

Specifies the type of device driver.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum DeviceType
```

Members

Member name	Description
Hardware	A hardware device. Using the flag to get direct access to the video hardware.
NullReference	A null device. This is a reference device that can do everything except render a scene.
Reference	A reference device. Use this flag to create a software emulated device.

See Also

Reference

[GraphicsAdapter.CheckDepthStencilMatch Method](#)

[GraphicsAdapter.CheckDeviceFormat Method](#)

[GraphicsAdapter.CheckDeviceFormatConversion Method](#)

[GraphicsAdapter.CheckDeviceMultiSampleType Method](#)

[GraphicsAdapter.CheckDeviceType Method](#)

[GraphicsAdapter.GetCapabilities Method](#)

[GraphicsAdapter.IsDeviceTypeAvailable Method](#)

[GraphicsDevice Constructor](#)

[GraphicsDeviceCapabilities.DeviceType Property](#)

[GraphicsDeviceCreationParameters Constructor](#)

[GraphicsDeviceCreationParameters.DeviceType Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode Structure

Describes the display mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct DisplayMode
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[DisplayMode](#)

[DisplayMode Members](#)







[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








DisplayMode Members

The following tables list the members exposed by the DisplayMode type.



Public Properties

	Name	Description
	AspectRatio	Gets the aspect ratio used by the graphics device.
	Format	Gets a value indicating the surface format of the display mode.
	Height	Gets a value indicating the screen height, in pixels.
	RefreshRate	Gets a value indicating the refresh rate. The value of 0 indicates an adapter default.
	TitleSafeArea	Returns the title safe area of the display.
	Width	Gets a value indicating the screen width, in pixels.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares two objects to determine whether they are the same.
	Op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[DisplayMode Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayMode Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DisplayMode Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayMode.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
DisplayMode.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
DisplayMode.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayMode.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [DisplayMode](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    DisplayMode left,  
    DisplayMode right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    DisplayMode left,  
    DisplayMode right  
)
```

Parameters

left

The object to the left of the inequality operator.

right

The object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[DisplayMode Structure](#)







[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode Properties

Public Properties

	Name	Description
	AspectRatio	Gets the aspect ratio used by the graphics device.
	Format	Gets a value indicating the surface format of the display mode.
	Height	Gets a value indicating the screen height, in pixels.
	RefreshRate	Gets a value indicating the refresh rate. The value of 0 indicates an adapter default.
	TitleSafeArea	Returns the title safe area of the display.
	Width	Gets a value indicating the screen width, in pixels.

See Also

Reference

[DisplayMode Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayMode.AspectRatio Property

Gets the aspect ratio used by the graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float AspectRatio { get; }
```

Property Value

Aspect ratio of the viewport.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.Format Property

Gets a value indicating the surface format of the display mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; }
```

Property Value

The surface format of the display mode.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.Height Property

Gets a value indicating the screen height, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

Screen height, in pixels.

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.RefreshRate Property

Gets a value indicating the refresh rate. The value of 0 indicates an adapter default.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int RefreshRate { get; }
```

Property Value

Refresh rate. The value of 0 indicates an adapter default.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.TitleSafeArea Property

Returns the title safe area of the display.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rectangle TitleSafeArea { get; }
```

Property Value

The title safe area.

Remarks

On television sets, text should not be displayed at the outer edges of the screen because it may not be visible to the user. The title safe area is the part of the screen where text should be displayed.

See Also

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayMode.Width Property

Gets a value indicating the screen width, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

Screen width, in pixels.

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[DisplayMode Structure](#)

[DisplayMode Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayModeCollection Structure

Manipulates a collection of [DisplayMode](#) structures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct DisplayModeCollection : IEnumerable<DisplayMode>
```

See Also

Reference

[DisplayModeCollection Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








DisplayModeCollection Members

The following tables list the members exposed by the DisplayModeCollection type.



Public Properties

	Name	Description
	Item	Retrieves the DisplayMode structure with the specified format.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetEnumerator	Gets an enumerator that can iterate through the DisplayModeCollection .
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also










Reference

[DisplayModeCollection Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayModeCollection Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetEnumerator	Gets an enumerator that can iterate through the DisplayModeCollection .
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
 	op_Equality	Compares two objects to determine whether they are the same.
 	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DisplayModeCollection Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayModeCollection.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
DisplayModeCollection.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
DisplayModeCollection.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[DisplayModeCollection Structure](#)

[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayModeCollection.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [DisplayModeCollection](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[DisplayModeCollection Structure](#)

[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayModeCollection.GetEnumerator Method

Gets an enumerator that can iterate through the [DisplayModeCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<DisplayMode> GetEnumerator ()
```

Return Value

Enumerator that can iterate through the [DisplayModeCollection](#).

See Also

Reference

[DisplayModeCollection Structure](#)

[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayModeCollection.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[DisplayModeCollection Structure](#)

[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayModeCollection.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    DisplayModeCollection left,  
    DisplayModeCollection right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[DisplayModeCollection Structure](#)

[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayModeCollection.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    DisplayModeCollection left,  
    DisplayModeCollection right  
)
```

Parameters

left

The object to the left of the inequality operator.

right

The object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[DisplayModeCollection Structure](#)


[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DisplayModeCollection Properties

Public Properties

	Name	Description
	Item	Retrieves the DisplayMode structure with the specified format.

See Also

Reference

[DisplayModeCollection Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DisplayModeCollection.Item Property

Retrieves the [DisplayMode](#) structure with the specified format.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerable<DisplayMode> this [
    SurfaceFormat format
] { get; }
```

Property Value

The [DisplayMode](#).

See Also

Reference

[DisplayModeCollection Structure](#)

[DisplayModeCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DriverInternalErrorException Class

The exception that is thrown when an internal driver error occurs. Applications should destroy and recreate the device when receiving this error.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class DriverInternalErrorException : ExternalException
```

See Also

Reference

[DriverInternalErrorException Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









DriverInternalErrorException Members

The following tables list the members exposed by the DriverInternalErrorException type.


Public Constructors

Name	Description
 DriverInternalErrorException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[DriverInternalErrorException Class](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

DriverInternalErrorException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
DriverInternalErrorException ()	Initializes a new instance of this class.
DriverInternalErrorException (String)	Initializes a new instance of this class with a specified error message.
DriverInternalErrorException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[DriverInternalErrorException Class](#)

[DriverInternalErrorException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DriverInternalErrorException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DriverInternalErrorException ()
```

Remarks

This constructor initializes the [Message](#) property of the new instance to a system-supplied message that describes the error, such as "Internal driver error." This message takes into account the current system culture.

The following table shows the initial property values for an instance of [DriverInternalErrorException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DriverInternalErrorException Class](#)

[DriverInternalErrorException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DriverInternalErrorException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DriverInternalErrorException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [DriverInternalErrorException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DriverInternalErrorException Class](#)

[DriverInternalErrorException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DriverInternalErrorException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DriverInternalErrorException (
    string message,
    Exception inner
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [DriverInternalErrorException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[DriverInternalErrorException Class](#)







[DriverInternalErrorException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

DriverInternalErrorException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[DriverInternalErrorException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

DriverInternalErrorException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[DriverInternalErrorException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer Class

Describes the rendering order of the vertices in a vertex buffer. Use **DynamicIndexBuffer** for storing indices for dynamic vertices and [IndexBuffer](#) for indices of non-dynamic arrays.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class DynamicIndexBuffer : IndexBuffer
```

Remarks

For more information on drawing with dynamic buffers, see [DynamicVertexBuffer Class](#).

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[DynamicIndexBuffer Members](#)

[Indices](#)

[DrawUserIndexedPrimitives](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista











DynamicIndexBuffer Members

The following tables list the members exposed by the DynamicIndexBuffer type.









Public Constructors

Name	Description
 DynamicIndexBuffer	Overloaded. Initializes a new instance of DynamicIndexBuffer.





Public Properties

Name	Description
 BufferUsage	(Inherited from IndexBuffer .)
 GraphicsDevice	(Inherited from GraphicsResource .)
 IndexElementSize	(Inherited from IndexBuffer .)
 IsContentLost	Determines if the index buffer data has been lost due to a lost device event.
 IsDisposed	(Inherited from GraphicsResource .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 SizeInBytes	(Inherited from IndexBuffer .)
 Tag	(Inherited from GraphicsResource .)



Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetData	(Inherited from IndexBuffer .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetData	Overloaded. Copies array data to the index buffer.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 ContentLost	Occurs when resources are lost due to a lost device event.
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicIndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer Constructor

Initializes a new instance of **DynamicIndexBuffer**.

Overload List

Name	Description
DynamicIndexBuffer (GraphicsDevice, Int32, BufferUsage, IndexElementSize)	Initializes a new instance of DynamicIndexBuffer with the specified parameters.
DynamicIndexBuffer (GraphicsDevice, Type, Int32, BufferUsage)	Initializes a new instance of DynamicIndexBuffer with the specified parameters.

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer Constructor (GraphicsDevice, Int32, BufferUsage, IndexElementSize)

Initializes a new instance of **DynamicIndexBuffer** with the specified parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DynamicIndexBuffer (
    GraphicsDevice graphicsDevice,
    int sizeInBytes,
    BufferUsage usage,
    IndexElementSize elementSize
)
```

Parameters

graphicsDevice

The associated graphics device of the index buffer.

sizeInBytes

The size, in bytes, of the index buffer.

usage

A set of options identifying the behaviors of this index buffer resource.

elementSize

The size, in bits, of an index element.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The <i>elementSize</i> parameter is invalid. Index buffers can only be created for indices which are sixteen or thirty-two bits in length. The <i>sizeInBytes</i> parameter is invalid. The resource size must be greater than zero. The <i>sizeInBytes</i> parameter is invalid. The total size of the index buffer must be an even multiple of the index element size (either 16 or 32 bits).
InvalidOperationException	This resource could not be created.

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer Constructor (GraphicsDevice, Type, Int32, BufferUsage)

Initializes a new instance of **DynamicIndexBuffer** with the specified parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DynamicIndexBuffer (
    GraphicsDevice graphicsDevice,
    Type indexType,
    int elementCount,
    BufferUsage usage
)
```

Parameters

graphicsDevice

The associated graphics device of the index buffer.

indexType

Type to use for index values.

elementCount

Number of indices in the buffer.

usage

A set of options identifying the behaviors of this index buffer resource.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	The <i>elementCount</i> parameter is invalid. It must be greater than zero, and the index must be sixteen or thirty-two bits in length. Index buffers can only be created for indices that are sixteen or thirty-two bits in length.
InvalidOperationException	This resource could not be created.

See Also

Reference

[DynamicIndexBuffer Class](#)









[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetData	(Inherited from IndexBuffer .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetData	Overloaded. Copies array data to the index buffer.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_ContentLost	Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicIndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
DynamicIndexBuffer.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.
DynamicIndexBuffer.Dispose ()	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer.raise_ContentLost Method

Note

This method is available only when developing for Windows.

Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void raise_ContentLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

DynamicIndexBuffer.SetData Method

Copies array data to the index buffer.

Overload List

Name	Description
DynamicIndexBuffer.SetData (Int32, T[], Int32, Int32, SetDataOptions)	Copies array data to the index buffer.
DynamicIndexBuffer.SetData (T[], Int32, Int32, SetDataOptions)	Copies array data to the index buffer.
DynamicIndexBuffer.SetData (Int32, T, Int32, Int32)	(Inherited from IndexBuffer .)
DynamicIndexBuffer.SetData (T)	(Inherited from IndexBuffer .)
DynamicIndexBuffer.SetData (T, Int32, Int32)	(Inherited from IndexBuffer .)

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer.SetData Generic Method (Int32, T[], Int32, Int32, SetDataOptions)

Copies array data to the index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    int offsetInBytes,  
    T[] data,  
    int startIndex,  
    int elementCount,  
    SetDataOptions options  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

offsetInBytes

Number of bytes into the index buffer where copying will start.

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

options

Specifies whether existing data in the buffer will be kept after this operation. [Discard](#) is not valid on Xbox 360, but dynamic geometry may be rendered on the Xbox 360 by using [DrawUserIndexedPrimitives](#) instead of setting the data for the index buffer.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer.SetData Generic Method (T[], Int32, Int32, SetDataOptions)

Copies array data to the index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    T[] data,  
    int startIndex,  
    int elementCount,  
    SetDataOptions options  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

options

Specifies whether existing data in the buffer will be kept after this operation. [Discard](#) is not valid on Xbox 360, but dynamic geometry may be rendered on the Xbox 360 by using [DrawUserIndexedPrimitives](#) instead of setting the data for the index buffer.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[DynamicIndexBuffer Class](#)











[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer Properties

Public Properties

	Name	Description
	BufferUsage	(Inherited from IndexBuffer .)
	GraphicsDevice	(Inherited from GraphicsResource .)
	IndexElementSize	(Inherited from IndexBuffer .)
	IsContentLost	Determines if the index buffer data has been lost due to a lost device event.
	IsDisposed	(Inherited from GraphicsResource .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	SizeInBytes	(Inherited from IndexBuffer .)
	Tag	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicIndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer.IsContentLost Property

Determines if the index buffer data has been lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsContentLost { get; }
```

Property Value

true if the content was lost; **false** otherwise.

See Also

Reference

[DynamicIndexBuffer Class](#)



[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicIndexBuffer Events

Public Events

	Name	Description
	ContentLost	Occurs when resources are lost due to a lost device event.
	Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicIndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicIndexBuffer.ContentLost Event

Occurs when resources are lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual event EventHandler ContentLost
```

Remarks

When this event occurs, the [IsContentLost](#) property, of the index buffer, is set to **true**.

See Also

Reference

[DynamicIndexBuffer Class](#)

[DynamicIndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer Class

Represents a list of 3D vertices to be streamed to the graphics device. Use **DynamicVertexBuffer** for dynamic vertex arrays and [VertexBuffer](#) for non-dynamic vertex arrays.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class DynamicVertexBuffer : VertexBuffer
```

Remarks

In situations where your game frequently modifies a vertex buffer, it is recommended that the buffer be instantiated or derived from **DynamicVertexBuffer** instead of the [VertexBuffer](#) class. **DynamicVertexBuffer** is optimized for frequent vertex data modification. However, to fully maximize the benefits of this class, your data must be restored after any occurrence of a [ContentLost](#) event. This event occurs whenever the related graphics device is lost. After resetting the graphics device and restoring any static resources, the [ContentLost](#) event handler is then called. Calling [SetData](#) inside this handler restores any dynamic resource data.

You can declare the event handler after constructing your dynamic vertex buffer. For example, the following code sample first constructs a dynamic vertex buffer (called `vertexBuffer`), declares a [ContentLost](#) event handler (called `vertexBuffer_ContentLost`). The final step initializes the data with a call to [SetData](#). This is necessary for correct restoration of any existing particles after a lost device event.

```
vertexBuffer = new DynamicVertexBuffer(GraphicsDevice, size, BufferUsage.Points);  
vertexBuffer.ContentLost += new EventHandler(vertexBuffer_ContentLost);  
vertexBuffer.SetData(particles);
```

See Also

Reference

[DynamicVertexBuffer Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista










DynamicVertexBuffer Members

The following tables list the members exposed by the DynamicVertexBuffer type.









Public Constructors

Name	Description
 DynamicVertexBuffer	Overloaded. Initializes a new instance of DynamicVertexBuffer.





Public Properties

Name	Description
 BufferUsage	(Inherited from VertexBuffer .)
 GraphicsDevice	(Inherited from GraphicsResource .)
 IsContentLost	Determines if the index buffer data has been lost due to a lost device event.
 IsDisposed	(Inherited from GraphicsResource .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 SizeInBytes	(Inherited from VertexBuffer .)
 Tag	(Inherited from GraphicsResource .)



Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetData	(Inherited from VertexBuffer .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetData	Overloaded. Copies array data to the vertex buffer.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 ContentLost	Occurs when resources are lost due to a lost device event.
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicVertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer Constructor

Initializes a new instance of **DynamicVertexBuffer**.

Overload List

Name	Description
DynamicVertexBuffer (GraphicsDevice, Int32, BufferUsage)	Initializes a new instance of DynamicVertexBuffer with the specified parameters.
DynamicVertexBuffer (GraphicsDevice, Type, Int32, BufferUsage)	Initializes a new instance of DynamicVertexBuffer with the specified parameters.

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer Constructor (GraphicsDevice, Int32, BufferUsage)

Initializes a new instance of **DynamicVertexBuffer** with the specified parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DynamicVertexBuffer (  
    GraphicsDevice graphicsDevice,  
    int sizeInBytes,  
    BufferUsage usage  
)
```

Parameters

graphicsDevice

The graphics device to associate with this vertex buffer.

sizeInBytes

The number of bytes to allocate for this vertex buffer resource.

usage

A set of options identifying the behaviors of this vertex buffer resource. It is good practice to match the *usage* parameter with the *createOptions* parameter in the [GraphicsDevice](#) constructor.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>graphicsDevice</i> must not be null when creating new resources.
ArgumentOutOfRangeException	<i>sizeInBytes</i> must be greater than zero.
InvalidOperationException	This resource could not be created.

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer Constructor (GraphicsDevice, Type, Int32, BufferUsage)

Initializes a new instance of **DynamicVertexBuffer** with the specified parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DynamicVertexBuffer (
    GraphicsDevice graphicsDevice,
    Type vertexType,
    int elementCount,
    BufferUsage usage
)
```

Parameters

graphicsDevice

The graphics device to associate with this vertex buffer.

vertexType

The type of vertices in this vertex buffer.

elementCount

The number of elements in this vertex buffer.

usage

A set of options identifying the behaviors of this vertex buffer resource. It is good practice to match the *usage* parameter with the *createOptions* parameter in the [GraphicsDevice](#) constructor.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>graphicsDevice</i> must not be null when creating new resources.
ArgumentOutOfRangeException	<i>elementCount</i> must be greater than zero.
InvalidOperationException	This resource could not be created.

See Also

Reference

[DynamicVertexBuffer Class](#)









[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetData	(Inherited from VertexBuffer .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetData	Overloaded. Copies array data to the vertex buffer.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_ContentLost	Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicVertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
DynamicVertexBuffer.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.
DynamicVertexBuffer.Dispose ()	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer.raise_ContentLost Method

Note

This method is available only when developing for Windows.

Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void raise_ContentLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

DynamicVertexBuffer.SetData Method

Copies array data to the vertex buffer.

Overload List

Name	Description
DynamicVertexBuffer.SetData (Int32, T[], Int32, Int32, Int32, SetDataOptions)	Copies array data to the vertex buffer.
DynamicVertexBuffer.SetData (T[], Int32, Int32, SetDataOptions)	Copies array data to the vertex buffer.
DynamicVertexBuffer.SetData (Int32, T, Int32, Int32, Int32)	(Inherited from VertexBuffer.)
DynamicVertexBuffer.SetData (T)	(Inherited from VertexBuffer.)
DynamicVertexBuffer.SetData (T, Int32, Int32)	(Inherited from VertexBuffer.)

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer.SetData Generic Method (Int32, T[], Int32, Int32, Int32, SetDataOptions)

Copies array data to the vertex buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    int offsetInBytes,  
    T[] data,  
    int startIndex,  
    int elementCount,  
    int vertexStride,  
    SetDataOptions options  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

offsetInBytes

Number of bytes into the index buffer where copying will start.

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

vertexStride

The size, in bytes, of the elements in the vertex buffer.

options

Specifies whether existing data in the buffer will be kept after this operation. [Discard](#) is not valid on Xbox 360, but dynamic geometry may be rendered on the Xbox 360 by using [DrawUserIndexedPrimitives](#) instead of setting the data for the vertex buffer.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer.SetData Generic Method (T[], Int32, Int32, SetDataOptions)

Copies array data to the vertex buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    T[] data,  
    int startIndex,  
    int elementCount,  
    SetDataOptions options  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

options

Specifies whether existing data in the buffer will be kept after this operation. [Discard](#) is not valid on Xbox 360, but dynamic geometry may be rendered on the Xbox 360 by using [DrawUserIndexedPrimitives](#) instead of setting the data for the vertex buffer.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[DynamicVertexBuffer Class](#)










[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer Properties

Public Properties

	Name	Description
	BufferUsage	(Inherited from VertexBuffer .)
	GraphicsDevice	(Inherited from GraphicsResource .)
	IsContentLost	Determines if the index buffer data has been lost due to a lost device event.
	IsDisposed	(Inherited from GraphicsResource .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	SizeInBytes	(Inherited from VertexBuffer .)
	Tag	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicVertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer.IsContentLost Property

Determines if the index buffer data has been lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsContentLost { get; }
```

Property Value

true if the content was lost; **false** otherwise.

See Also

Reference

[DynamicVertexBuffer Class](#)



[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

DynamicVertexBuffer Events

Public Events

	Name	Description
	ContentLost	Occurs when resources are lost due to a lost device event.
	Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[DynamicVertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

DynamicVertexBuffer.ContentLost Event

Occurs when resources are lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual event EventHandler ContentLost
```

When this event occurs, the [IsContentLost](#) property, of the vertex buffer, is set to **true**.

See Also

Reference

[DynamicVertexBuffer Class](#)

[DynamicVertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Class

Used to set and query effects and choose techniques.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class Effect : IDisposable
```

See Also

Concepts

[Shader Content Catalog at XNA Creators Club Online](#)

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista









Effect Members

The following tables list the members exposed by the Effect type.














Public Constructors

	Name	Description
	Effect	Overloaded. Initializes a new instance of this class.






Public Properties

	Name	Description
	Creator	Gets the name of the effect creator.
	CurrentTechnique	Gets or sets the active technique.
	EffectPool	Gets an EffectPool representing the pool of shared parameters.
	Functions	Gets a collection of functions that can render the effect.
	GraphicsDevice	Gets the graphics device that created the effect.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Parameters	Gets a collection of parameters used for this effect.
	Techniques	Gets a collection of techniques that are defined for this effect.




Public Methods

	Name	Description
	Begin	Overloaded. Begins application of the active technique.
	Clone	Creates a clone of an effect.
	CommitChanges	Propagates the state change that occurs inside of an active pass to the device before rendering.
	CompileEffectFromFile	Overloaded. Compiles an effect from a file or stream containing the effect source code.
	CompileEffectFromSource	Compiles an effect from a string containing the effect source code.
	Disassemble	Overloaded. Disassembles an effect.
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	End	Ends the application of the current technique.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.
	raise_Lost	Raises an Lost event when called from within a derived class.
	raise_Reset	Raises an Reset event when called from within a derived class.

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).
	Lost	Occurs when an object is lost, normally just before a device is reset.
	Reset	Occurs after the device is reset.

See Also

Reference

[Effect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect Constructor

Initializes a new instance of this class.

Overload List

Name	Description
Effect (GraphicsDevice, Byte[], CompilerOptions, EffectPool)	Initializes a new instance of this class from a compiled effect, specifying the compiled effect as a byte array.
Effect (GraphicsDevice, Effect)	Initializes a new instance of this class, specifying the graphics device to associate with this effect and the effect to clone.
Effect (GraphicsDevice, Stream, CompilerOptions, EffectPool)	Initializes a new instance of this class from a compiled effect, specifying the compiled effect as a stream.
Effect (GraphicsDevice, Stream, Int32, CompilerOptions, EffectPool)	Initializes a new instance of this class from a compiled effect, specifying the compiled effect as a stream, as well as the number of bytes in the stream.
Effect (GraphicsDevice, String, CompilerOptions, EffectPool)	Initializes a new instance of this class from a compiled effect, specifying the compiled effect by file name.

Tip

In most cases, it is preferable to initialize a new instance of an effect using the [ContentManager.Load<Effect>](#) method rather than using the **Effect** constructors. For an example of initializing an effect with this method, please see [How To: Create and Apply Custom Effects](#).

See Also

Concepts

[Shader Content Catalog at XNA Creators Club Online](#)

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect Constructor (GraphicsDevice, Byte[], CompilerOptions, EffectPool)

Initializes a new instance of this class from a compiled effect, specifying the compiled effect as a byte array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Effect (
    GraphicsDevice graphicsDevice,
    byte[] effectCode,
    CompilerOptions options,
    EffectPool pool
)
```

Parameters

graphicsDevice

The graphics device that will create the effect.

effectCode

Byte array containing the compiled byte code.

options

Compilation optimization options.

pool

Specifies a pool of resources to share between effects.

Exceptions

Exception type	Condition
ArgumentNullException	<i>effectCode</i> or <i>graphicsDevice</i> is null .
InvalidOperationException	One of the following conditions is true: <ul style="list-style-type: none"> The code contained in <i>effectCode</i> is not compiled. Creating an effect requires that the shader code has been compiled. The code contained in <i>effectCode</i> does not have a size that is a multiple of four bytes. Effect and shader code must have a size that is a multiple of four bytes. Unable to create the effect resource on the graphics device.
ObjectDisposedException	Effect was called after <i>pool</i> was disposed.

Remarks

Tip

In most cases, it is preferable to initialize a new instance of an effect using the [ContentManager.Load<Effect>](#) method rather than using the **Effect** constructors. For an example of initializing an effect with this method, see [How To: Create and Apply Custom Effects](#).

Obtaining the compiled byte code for *effectCode* is a two-part process. First, you must compile the effect by using the static method [Effect.CompileEffectFromFile](#) or [Effect.CompileEffectFromSource](#), either of which returns an instance of [CompiledEffect](#). [CompiledEffect.GetEffectCode](#) returns the compiled byte code from the compiled effect.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Constructor (GraphicsDevice, Effect)

Initializes a new instance of this class, specifying the graphics device to associate with this effect and the effect to clone.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected Effect (
    GraphicsDevice graphicsDevice,
    Effect cloneSource
)
```

Parameters

graphicsDevice

The graphics device that will create the effect.

cloneSource

The effect to clone.

Exceptions

Exception type	Condition
ArgumentNullException	<i>cloneSource</i> or <i>graphicsDevice</i> is null .
InvalidOperationException	Unable to create the effect resource on the graphics device.
ObjectDisposedException	Effect was called after <i>cloneSource</i> was disposed.

Tip

In most cases, it is preferable to initialize a new instance of an effect using the [ContentManager.Load<Effect>](#) method rather than using the **Effect** constructors. For an example of initializing an effect by using this method, see [How To: Create and Apply Custom Effects](#).

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Constructor (GraphicsDevice, Stream, CompilerOptions, EffectPool)

Initializes a new instance of this class from a compiled effect, specifying the compiled effect as a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Effect (
    GraphicsDevice graphicsDevice,
    Stream effectCodeFileStream,
    CompilerOptions options,
    EffectPool pool
)
```

Parameters

graphicsDevice

The graphics device that will create the effect.

effectCodeFileStream

Stream containing the compiled byte code.

options

Compilation optimization options.

pool

Specifies a pool of resources to share between effects.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null . The graphics device must not be null when creating new resources.
ObjectDisposedException	Effect was called after <i>pool</i> was disposed.
InvalidOperationException	<ul style="list-style-type: none"> The code contained in <i>effectCodeFileStream</i> is not compiled. Creating an effect requires that the shader code has been compiled. The code contained in <i>effectCodeFileStream</i> does not have a size that is a multiple of four bytes. Effect and shader code must have a size that is a multiple of four bytes. Unable to create the effect resource on the graphics device.

Remarks

Tip

In most cases, it is preferable to initialize a new instance of an effect using the [ContentManager.Load<Effect>](#) method rather than using the **Effect** constructors. For an example of initializing an effect with this method, see [How To: Create and Apply Custom Effects](#).

Obtaining the compiled byte code for *effectCodeFileStream* is a two-part process. First, you must compile the effect by using the static method [Effect.CompileEffectFromFile](#) or [Effect.CompileEffectFromSource](#), either of which returns an instance of [CompiledEffect](#). [CompiledEffect.GetEffectCode](#) returns the compiled byte code from the compiled effect.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Constructor (GraphicsDevice, Stream, Int32, CompilerOptions, EffectPool)

Initializes a new instance of this class from a compiled effect, specifying the compiled effect as a stream, as well as the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Effect (
    GraphicsDevice graphicsDevice,
    Stream effectCodeFileStream,
    int numberBytes,
    CompilerOptions options,
    EffectPool pool
)
```

Parameters

graphicsDevice

The graphics device that will create the effect.

effectCodeFileStream

Stream containing the compiled byte code.

numberBytes

The number of bytes to read from the file.

options

Compilation optimization options.

pool

Specifies a pool of resources to share between effects.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null . The graphics device must not be null when creating new resources.
InvalidOperationException	One of the following conditions is true: <ul style="list-style-type: none"> <i>effectCodeFileStream</i> does not contain enough data to accommodate the number of bytes requested by the <i>numberBytes</i> parameter. The code contained in <i>effectCodeFileStream</i> is not compiled. Creating an effect requires that the shader code has been compiled. The code contained in <i>effectCodeFileStream</i> does not have a size that is a multiple of four bytes. Effect and shader code must have a size that is a multiple of four bytes. Unable to create the effect resource on the graphics device.
ObjectDisposedException	Effect was called after <i>pool</i> was disposed.

Remarks

Tip

In most cases, it is preferable to initialize a new instance of an effect using the [ContentManager.Load<Effect>](#) method rather than using the **Effect** constructors. For an example of initializing an effect with this method, see [How To: Create and Apply Custom Effects](#).

Obtaining the compiled byte code for *effectCodeFileStream* is a two-part process. First, you must compile the effect by using the static method [Effect.CompileEffectFromFile](#) or [Effect.CompileEffectFromSource](#), either of which returns an instance of [CompiledEffect](#). [CompiledEffect.GetEffectCode](#) returns the compiled byte code from the compiled effect.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

[Platforms Xbox 360, Windows XP SP2, Windows Vista](#)

Effect Constructor (GraphicsDevice, String, CompilerOptions, EffectPool)

Initializes a new instance of this class from a compiled effect, specifying the compiled effect by file name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Effect (
    GraphicsDevice graphicsDevice,
    string effectCodeFile,
    CompilerOptions options,
    EffectPool pool
)
```

Parameters

graphicsDevice

The graphics device that will create the effect.

effectCodeFile

A file containing a compiled byte code.

options

Compilation optimization options.

pool

Specifies a pool of resources to share between effects.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null . The graphics device must not be null when creating new resources.
ObjectDisposedException	Effect was called after <i>pool</i> was disposed.
InvalidOperationException	<ul style="list-style-type: none"> The code contained in <i>effectCodeFile</i> is not compiled. Creating an effect requires that the shader code has been compiled. The code contained in <i>effectCodeFile</i> does not have a size that is a multiple of four bytes. Effect and shader code must have a size that is a multiple of four bytes. Unable to create the effect resource on the graphics device.

Tip

In most cases, it is preferable to initialize a new instance of an effect using the [ContentManager.Load<Effect>](#) method rather than using the **Effect** constructors. For an example of initializing an effect with this method, please see [How To: Create and Apply Custom Effects](#).

Remarks Obtaining the compiled byte code for *effectCodeFile* is a two-part process. First, the effect must be compiled using the static methods [Effect.CompileEffectFromFile](#) or [Effect.CompileEffectFromSource](#), which returns an instance of [CompiledEffect](#). [CompiledEffect.GetEffectCode](#) returns the compiled byte code from the compiled effect.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[Effect.CompileEffectFromFile Method](#)

[Effect.CompileEffectFromSource Method](#)

[Effect Class](#)
















[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)






Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Methods

Public Methods

	Name	Description
	Begin	Overloaded. Begins application of the active technique.
	Clone	Creates a clone of an effect.
	CommitChanges	Propagates the state change that occurs inside of an active pass to the device before rendering.
 	CompileEffectFromFile	Overloaded. Compiles an effect from a file or stream containing the effect source code.
 	CompileEffectFromSource	Compiles an effect from a string containing the effect source code.
	Disassemble	Overloaded. Disassembles an effect.
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	End	Ends the application of the current technique.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.
	raise_Lost	Raises an Lost event when called from within a derived class.
	raise_Reset	Raises an Reset event when called from within a derived class.

See Also

Reference

[Effect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.Begin Method

Begins application of the active technique.

Overload List

Name	Description
Effect.Begin ()	Begins application of the active technique.
Effect.Begin (SaveStateMode)	Begins application of the active technique, specifying options for saving state.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.Begin Method ()

Begins application of the active technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Begin Method (SaveStateMode)

Begins application of the active technique, specifying options for saving state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin (  
    SaveStateMode saveStateMode  
)
```

Parameters

saveStateMode

Options for saving the state prior to application of the technique.

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Clone Method

Creates a clone of an effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual Effect Clone (  
    GraphicsDevice device  
)
```

Parameters

device

The device associated with the effect.

Return Value

The cloned effect.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.CommitChanges Method

Propagates the state change that occurs inside of an active pass to the device before rendering.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void CommitChanges ()
```

RemarksIf the application changes any effect state using the [Effect](#) class members that modify state inside of an [Begin/End](#) matching pair, the application must call [CommitChanges](#) before any of the various [DrawPrimitives](#) methods are called to propagate state changes to the device before rendering. If no state changes occur within an [Begin/End](#) matching pair, it is not necessary to call [CommitChanges](#).

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

Effect.CompileEffectFromFile Method

Compiles an effect from a file or stream containing the effect source code.

Overload List

Name	Description
Effect.CompileEffectFromFile (Stream, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)	Compiles an effect from a stream containing the effect source code.
Effect.CompileEffectFromFile (Stream, Int32, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)	Compiles an effect from a stream containing the effect source code, specifying the number of bytes in the stream.
Effect.CompileEffectFromFile (String, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)	Compiles an effect from a file containing the effect source code.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.CompileEffectFromFile Method (Stream, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles an effect from a stream containing the effect source code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledEffect CompileEffectFromFile (
    Stream effectFileStream,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    TargetPlatform platform
)
```

Parameters

effectFileStream

File stream containing the effect source code.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

The target platform for the compilation.

Return Value

The compiled effect.

Exceptions

Exception type	Condition
ArgumentNullException	<i>effectFileStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>effectFileStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>effectFileStream</i> does not contain enough data to support this call.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.CompileEffectFromFile Method (Stream, Int32, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles an effect from a stream containing the effect source code, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledEffect CompileEffectFromFile (
    Stream effectFileStream,
    int numberBytes,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    TargetPlatform platform
)
```

Parameters

effectFileStream

File stream containing the effect source code.

numberBytes

The number of bytes in *effectFileStream*.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

The target platform for the compilation.

Return Value

The compiled effect.

Exceptions

Exception type	Condition
ArgumentNullException	<i>effectFileStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>effectFileStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>effectFileStream</i> does not contain enough data to support this call.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.CompileEffectFromFile Method (String, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles an effect from a file containing the effect source code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledEffect CompileEffectFromFile (
    string effectFile,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    TargetPlatform platform
)
```

Parameters

effectFile

The name of the file containing the effect source code.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

The target platform for the compilation.

Return Value

The compiled effect.

Exceptions

Exception type	Condition
ArgumentException	<i>effectFile</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>effectFile</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>effectFile</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>effectFile</i> was not found.
NotSupportedException	<i>effectFile</i> is in an invalid format.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.CompileEffectFromSource Method

Note

This method is available only when developing for Windows.

Compiles an effect from a string containing the effect source code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledEffect CompileEffectFromSource (  
    string effectFileSource,  
    CompilerMacro[] preprocessorDefines,  
    CompilerIncludeHandler includeHandler,  
    CompilerOptions options,  
    TargetPlatform platform  
)
```

Parameters

effectFileSource

String containing the effect source code.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

The target platform for the compilation.

Return Value

The compiled effect.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.Disassemble Method

Disassembles an effect.

Overload List

Name	Description
Effect.Disassemble (Boolean)	Disassembles this effect.
Effect.Disassemble (Effect, Boolean)	Disassembles an effect.

Exceptions

Exception type	Condition
InsufficientMemoryException	Microsoft Direct3D could not allocate sufficient memory to complete the call.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.Disassemble Method (Boolean)

Note

This method is available only when developing for Windows.

Disassembles this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Disassemble (  
    bool enableColorCode  
)
```

Parameters

enableColorCode

[[MarshalAsAttribute\(U1\)](#)] **true** to enable color coding to make the disassembly easier to read.

Return Value

A string that contains the effect assembly (ASM).

Exceptions

Exception type	Condition
InsufficientMemoryException	Microsoft Direct3D could not allocate sufficient memory to complete the call.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.Disassemble Method (Effect, Boolean)

Note

This method is available only when developing for Windows.

Disassembles an effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static string Disassemble (  
    Effect effect,  
    bool enableColorCode  
)
```

Parameters

effect

The effect to disassemble.

enableColorCode

[[MarshalAsAttribute](#)(U1)] **true** to enable color coding to make the disassembly easier to read.

Return Value

A string that contains the effect assembly (ASM).

Exceptions

Exception type	Condition
InsufficientMemoryException	Microsoft Direct3D could not allocate sufficient memory to complete the call.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
Effect.Dispose ()	Immediately releases the unmanaged resources used by this object.
Effect.Dispose (Boolean)	Releases the unmanaged resources used by the Effect and optionally releases the managed resources.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Dispose Method (Boolean)

Releases the unmanaged resources used by the [Effect](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute](#)(U1)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [Effect](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.End Method

Ends the application of the current technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void End ()
```

Exceptions

Exception type	Condition
InvalidOperationException	End was called, but Begin has not yet been called. Begin must be called successfully before End can be called.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

[How To: Use BasicEffect](#)

[How To: Create Custom Texture Effects](#)

Reference

[Effect.Begin Method](#)

[Effect.CurrentTechnique Property](#)

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Effect.raise_Lost Method

Note

This method is available only when developing for Windows.

Raises an [Lost](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Lost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass into the event handler.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.raise_Reset Method

Note

This method is available only when developing for Windows.

Raises an [Reset](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Reset (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass into the event handler.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Effect.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[Effect Class](#)









[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Properties

Public Properties

	Name	Description
	Creator	Gets the name of the effect creator.
	CurrentTechnique	Gets or sets the active technique.
	EffectPool	Gets an EffectPool representing the pool of shared parameters.
	Functions	Gets a collection of functions that can render the effect.
	GraphicsDevice	Gets the graphics device that created the effect.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Parameters	Gets a collection of parameters used for this effect.
	Techniques	Gets a collection of techniques that are defined for this effect.

See Also

Reference

[Effect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.Creator Property

Gets the name of the effect creator.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Creator { get; }
```

Property Value

The name of the effect creator.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.CurrentTechnique Property

Gets or sets the active technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectTechnique CurrentTechnique { get; set; }
```

Property Value

The current technique.

Exceptions

Exception type	Condition
ArgumentNullException	<i>CurrentTechnique</i> is null . <i>CurrentTechnique</i> cannot be null .

Remarks

If there are multiple techniques in an effect and you want to use a new technique in the next pass, you must set

CurrentTechnique to the new technique before making the rendering pass. It is not enough to call the [EffectPass.Begin](#) method of the next technique.

Note

The [BasicEffect](#) class, which is derived from [Effect](#) and inherits the [Techniques](#) collection, contains only one available technique to set as the active technique. The active technique applies the settings that have been set as the properties of the [BasicEffect](#).

Example

```
effect.CurrentTechnique = effect.Techniques["TransformTechnique"];
```

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

[How To: Create Custom Texture Effects](#)

Reference

[Effect.Techniques Property](#)

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.EffectPool Property

Gets an [EffectPool](#) representing the pool of shared parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectPool EffectPool { get; }
```

Property Value

The pool of shared parameters.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Functions Property

Gets a collection of functions that can render the effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectFunctionCollection Functions { get; }
```

Property Value

Collection of functions that can render the effect.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.GraphicsDevice Property

Gets the graphics device that created the effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The graphics device that created the effect.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Parameters Property

Gets a collection of parameters used for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterCollection Parameters { get; }
```

Property Value

The collection of parameters used for this effect.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

[How To: Create Custom Texture Effects](#)

[How To: Use EffectParameters and EffectTechniques](#)

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Techniques Property

Gets a collection of techniques that are defined for this effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectTechniqueCollection Techniques { get; }
```

Property Value

A collection of techniques that are defined for this effect.

Remarks

Note

The [BasicEffect](#) class, which is derived from [Effect](#) and inherits the [Techniques](#) collection, contains only one available technique which will apply the settings which have been set as the properties of the [BasicEffect](#).

Example In this example, an effect file is shown that defines a single technique called *TransformTechnique*.

```
uniform extern float4x4 WorldViewProj : WORLDVIEWPROJECTION;

struct VS_OUTPUT
{
    float4 position : POSITION;
    float4 color : COLOR0;
};

VS_OUTPUT Transform(
    float4 Pos : POSITION,
    float4 Color : COLOR0 )
{
    VS_OUTPUT Out = (VS_OUTPUT)0;

    Out.position = mul(Pos, WorldViewProj);
    Out.color = Color;

    return Out;
}

float4 PixelShader( VS_OUTPUT vsout ) : COLOR
{
    return vsout.color;
}

technique TransformTechnique
{
    pass P0
    {
        vertexShader = compile vs_2_0 Transform();
        pixelShader = compile ps_1_1 PixelShader();
    }
}
```

When this effect file is loaded as an [Effect](#) by the [ContentManager](#), this technique may be set as the active technique by setting [CurrentTechnique](#) to `effect.Techniques["TransformTechnique"]`.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

[How To: Use EffectParameters and EffectTechniques](#)

Reference

[Effect.CurrentTechnique Property](#)

[Effect Class](#)




[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).
	Lost	Occurs when an object is lost, normally just before a device is reset.
	Reset	Occurs after the device is reset.

See Also

Reference

[Effect Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Effect.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

Remarks [IsDisposed](#) indicates whether that object has been disposed.

Example

To add an event handler that listens for the **Disposing** event, use the following C# code.

C#

```
obj.Disposing += new System.EventHandler( this.OnDisposing );
```

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Lost Event

Occurs when an object is lost, normally just before a device is reset.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Lost
```

Example

To add an event handler that listens for a **Lost** event, use the following C# code.

C#

```
effect.Lost += new System.EventHandler( this.LostEventHandler );
```

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Effect.Reset Event

Occurs after the device is reset.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Reset
```

Example

To add an event handler that listens for a **Reset** event, use the following C# code.

C#

```
effect.Reset += new System.EventHandler( this.ResetEventHandler );
```

See Also

Reference

[Effect Class](#)

[Effect Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation Class

Represents an annotation to an [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectAnnotation
```

See Also

Reference

[EffectAnnotation Members](#)







[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista














EffectAnnotation Members

The following tables list the members exposed by the EffectAnnotation type.



Public Properties

Name	Description
 ColumnCount	Gets the number of columns in this effect annotation.
 Name	Gets the name of the effect annotation.
 ParameterClass	Gets the parameter class of this effect annotation.
 ParameterType	Gets the parameter type of this effect annotation.
 RowCount	Gets the row count of this effect annotation.
 Semantic	Gets the semantic of this effect annotation.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetValueBoolean	Gets the value of the EffectAnnotation as a Boolean.
 GetValueInt32	Gets the value of the EffectAnnotation as a Int32.
 GetValueMatrix	Gets the value of the EffectAnnotation as a Int32.
 GetValueSingle	Gets the value of the EffectAnnotation as a Single.
 GetValueString	Gets the value of the EffectAnnotation as a String.
 GetValueVector2	Gets the value of the EffectAnnotation as a Vector2 .
 GetValueVector3	Gets the value of the EffectAnnotation as a Vector3 .
 GetValueVector4	Gets the value of the EffectAnnotation as a Vector4 .
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also














Reference

[EffectAnnotation Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotation Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetValueBoolean	Gets the value of the EffectAnnotation as a Boolean.
	GetValueInt32	Gets the value of the EffectAnnotation as a Int32.
	GetValueMatrix	Gets the value of the EffectAnnotation as a Int32.
	GetValueSingle	Gets the value of the EffectAnnotation as a Single.
	GetValueString	Gets the value of the EffectAnnotation as a String.
	GetValueVector2	Gets the value of the EffectAnnotation as a Vector2 .
	GetValueVector3	Gets the value of the EffectAnnotation as a Vector3 .
	GetValueVector4	Gets the value of the EffectAnnotation as a Vector4 .
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectAnnotation Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotation.GetValueBoolean Method

Gets the value of the [EffectAnnotation](#) as a [Boolean](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool GetValueBoolean ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Boolean](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueInt32 Method

Gets the value of the [EffectAnnotation](#) as a [Int32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int GetValueInt32 ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Int32](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueMatrix Method

Gets the value of the [EffectAnnotation](#) as a [Int32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix GetValueMatrix ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Int32](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueSingle Method

Gets the value of the [EffectAnnotation](#) as a [Single](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GetValueSingle ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Single](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueString Method

Gets the value of the [EffectAnnotation](#) as a [String](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string GetValueString ()
```

Return Value

The value of the [EffectAnnotation](#) as a [String](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueVector2 Method

Gets the value of the [EffectAnnotation](#) as a [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 GetValueVector2 ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Vector2](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueVector3 Method

Gets the value of the [EffectAnnotation](#) as a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 GetValueVector3 ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Vector3](#).

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.GetValueVector4 Method

Gets the value of the [EffectAnnotation](#) as a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 GetValueVector4 ()
```

Return Value

The value of the [EffectAnnotation](#) as a [Vector4](#).

See Also

Reference

[EffectAnnotation Class](#)







[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation Properties

Public Properties

	Name	Description
	ColumnCount	Gets the number of columns in this effect annotation.
	Name	Gets the name of the effect annotation.
	ParameterClass	Gets the parameter class of this effect annotation.
	ParameterType	Gets the parameter type of this effect annotation.
	RowCount	Gets the row count of this effect annotation.
	Semantic	Gets the semantic of this effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotation.ColumnCount Property

Gets the number of columns in this effect annotation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ColumnCount { get; }
```

Property Value

The number of columns in this effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.Name Property

Gets the name of the effect annotation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of the effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.ParameterClass Property

Gets the parameter class of this effect annotation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterClass ParameterClass { get; }
```

Property Value

The parameter class of this effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.ParameterType Property

Gets the parameter type of this effect annotation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterType ParameterType { get; }
```

Property Value

The parameter type of this effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.RowCount Property

Gets the row count of this effect annotation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int RowCount { get; }
```

Property Value

The row count of this effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotation.Semantic Property

Gets the semantic of this effect annotation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Semantic { get; }
```

Property Value

The semantic of this effect annotation.

See Also

Reference

[EffectAnnotation Class](#)

[EffectAnnotation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotationCollection Class

Manipulates a collection of [EffectAnnotation](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectAnnotationCollection : IEnumerable<EffectAnnotation>
```

See Also

Reference

[EffectAnnotationCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista







EffectAnnotationCollection Members

The following tables list the members exposed by the EffectAnnotationCollection type.



Public Properties

	Name	Description
	Count	Gets the number of EffectAnnotation objects in this EffectAnnotationCollection .
	Item	Overloaded. Gets an EffectAnnotation object.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the EffectAnnotationCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[EffectAnnotationCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotationCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the EffectAnnotationCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectAnnotationCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotationCollection.GetEnumerator Method

Returns an enumerator that can iterate through the [EffectAnnotationCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<EffectAnnotation> GetEnumerator ()
```

Return Value

The iterator.

See Also

Reference

[EffectAnnotationCollection Class](#)



[EffectAnnotationCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotationCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of EffectAnnotation objects in this EffectAnnotationCollection .
	Item	Overloaded. Gets an EffectAnnotation object.

See Also

Reference

[EffectAnnotationCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotationCollection.Count Property

Gets the number of [EffectAnnotation](#) objects in this [EffectAnnotationCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of [EffectAnnotation](#) objects in this [EffectAnnotationCollection](#).

See Also

Reference

[EffectAnnotationCollection Class](#)

[EffectAnnotationCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotationCollection.Item Property

Gets an [EffectAnnotation](#) object.

Overload List

Name	Description
EffectAnnotationCollection.Item (Int32)	Gets a specific EffectAnnotation object by using an index value.
EffectAnnotationCollection.Item (String)	Gets a specific EffectAnnotation object by using a name.

See Also

Reference

[EffectAnnotationCollection Class](#)

[EffectAnnotationCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectAnnotationCollection.Item Property (Int32)

Gets a specific [EffectAnnotation](#) object by using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectAnnotation this [
    int index
] { get; }
```

Property Value

The [EffectAnnotation](#) object at index *index*.

See Also

Reference

[EffectAnnotationCollection Class](#)

[EffectAnnotationCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectAnnotationCollection.Item Property (String)

Gets a specific [EffectAnnotation](#) object by using a name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectAnnotation this [
    string name
] { get; }
```

Property Value

The [EffectAnnotation](#) object named *name*.

See Also

Reference

[EffectAnnotationCollection Class](#)

[EffectAnnotationCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectFunction Class

Represents a function on an [Effect](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectFunction
```

See Also

Reference

[EffectFunction Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






EffectFunction Members

The following tables list the members exposed by the EffectFunction type.



Public Properties

	Name	Description
	Name	Gets the name of the function.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[EffectFunction Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunction Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[EffectFunction Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunction Properties

Public Properties

	Name	Description
	Name	Gets the name of the function.

See Also

Reference

[EffectFunction Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunction.Name Property

Gets the name of the function.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of the function.

See Also

Reference

[EffectFunction Class](#)

[EffectFunction Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectFunctionCollection Class

Manipulates a collection of [EffectFunction](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectFunctionCollection : IEnumerable<EffectFunction>
```

See Also

Reference

[EffectFunctionCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista







EffectFunctionCollection Members

The following tables list the members exposed by the EffectFunctionCollection type.



Public Properties

	Name	Description
	Count	Gets the number of EffectFunction objects in this EffectFunctionCollection .
	Item	Overloaded. Gets an EffectAnnotation object.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the EffectFunctionCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[EffectFunctionCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunctionCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the EffectFunctionCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectFunctionCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunctionCollection.GetEnumerator Method

Returns an enumerator that can iterate through the [EffectFunctionCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<EffectFunction> GetEnumerator ()
```

Return Value

The iterator.

See Also

Reference

[EffectFunctionCollection Class](#)



[EffectFunctionCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectFunctionCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of EffectFunction objects in this EffectFunctionCollection .
	Item	Overloaded. Gets an EffectAnnotation object.

See Also

Reference

[EffectFunctionCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunctionCollection.Count Property

Gets the number of [EffectFunction](#) objects in this [EffectFunctionCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of [EffectFunction](#) objects in this [EffectFunctionCollection](#).

See Also

Reference

[EffectFunctionCollection Class](#)

[EffectFunctionCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectFunctionCollection.Item Property

Gets an [EffectAnnotation](#) object.

Overload List

Name	Description
EffectFunctionCollection.Item (Int32)	Gets a specific EffectFunction object by using an index value.
EffectFunctionCollection.Item (String)	Gets a specific EffectFunction object by using a name.

See Also

Reference

[EffectFunctionCollection Class](#)

[EffectFunctionCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectFunctionCollection.Item Property (Int32)

Gets a specific [EffectFunction](#) object by using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectFunction this [
    int index
] { get; }
```

Property Value

The [EffectFunction](#) object at index *index*.

See Also

Reference

[EffectFunctionCollection Class](#)

[EffectFunctionCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectFunctionCollection.Item Property (String)

Gets a specific [EffectFunction](#) object by using a name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectFunction this [
    string name
] { get; }
```

Property Value

The [EffectFunction](#) object named *name*.

See Also

Reference

[EffectFunctionCollection Class](#)

[EffectFunctionCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter Class

Represents an [Effect](#) parameter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectParameter
```

Remarks Creating and assigning a **EffectParameter** instance for each technique in your [Effect](#) is significantly faster than using the [Parameters](#) indexed property on [Effect](#).

Example

To use a **EffectParameter** you must:

1. Create a **EffectParameter** for each parameter in your [Effect](#) that you will be setting in [Draw](#) or [Update](#).

C#

```
public EffectParameter mWorld;
public EffectParameter mCameraView;
public EffectParameter CameraPos;
public EffectParameter mCameraProj;
```

2. Assign an [Effect](#) parameter to your **EffectParameter**.

C#

```
mWorld = effect.Parameters["g_mWorld"];
mCameraView = effect.Parameters["g_mCameraView"];
CameraPos = effect.Parameters["g_CameraPos"];
mCameraProj = effect.Parameters["g_mCameraProj"];
```

3. Call [SetValue](#) on your **EffectParameter** to change the parameter value.

C#

```
MyEffect.CameraPos.SetValue(CameraPos);
MyEffect.mCameraView.SetValue(view);
MyEffect.mCameraProj.SetValue(projection);
MyEffect.LightPos.SetValue(LightPos);
MyEffect.mLightView.SetValue(Matrix.CreateLookAt(LightPos,
    bounds.Center, Vector3.Up));
```

See Also

Tasks

[How To: Use EffectParameters and EffectTechniques](#)

Reference

[EffectParameter Members](#)

[Effect.Parameters](#)










[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter Members

The following tables list the members exposed by the EffectParameter type.



Public Properties

Name	Description
 Annotations	Gets the collection of EffectAnnotation objects for this parameter.
 ColumnCount	Gets the number of columns in the parameter description.
 Elements	Gets the collection of effect parameters.
 Name	Gets the name of the parameter.
 ParameterClass	Gets the class of the parameter.
 ParameterType	Gets the type of the parameter.
 RowCount	Gets the number of rows in the parameter description.
 Semantic	Gets the semantic meaning, or usage, of the parameter.
 StructureMembers	Gets the collection of structure members.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetValueBoolean	Gets the value of the EffectParameter as a Boolean.
 GetValueBooleanArray	Gets the value of the EffectParameter as an array of Boolean.
 GetValueInt32	Gets the value of the EffectParameter as an Int32.
 GetValueInt32Array	Gets the value of the EffectParameter as an array of Int32.
 GetValueMatrix	Gets the value of the EffectParameter as a Matrix .
 GetValueMatrixArray	Gets the value of the EffectParameter as an array of Matrix .
 GetValueMatrixTranspose	Gets the value of the EffectParameter as a Matrix transpose.
 GetValueMatrixTransposeArray	Gets the value of the EffectParameter as an array of Matrix transpose.
 GetValueQuaternion	Gets the value of the EffectParameter as a Quaternion .
 GetValueQuaternionArray	Gets the value of the EffectParameter as an array of Quaternion .
 GetValueSingle	Gets the value of the EffectParameter as a Single.
 GetValueSingleArray	Gets the value of the EffectParameter as an array of Single.
 GetValueString	Gets the value of the EffectParameter as a String.
 GetValueTexture2D	Gets the value of the EffectParameter as a Texture2D .
 GetValueTexture3D	Gets the value of the EffectParameter as a Texture3D .
 GetValueTextureCube	Gets the value of the EffectParameter as a TextureCube .
 GetValueVector2	Gets the value of the EffectParameter as a Vector2 .
 GetValueVector2Array	Gets the value of the EffectParameter as an array of Vector2 .
 GetValueVector3	Gets the value of the EffectParameter as a Vector3 .
 GetValueVector3Array	Gets the value of the EffectParameter as an array of Vector3 .
 GetValueVector4	Gets the value of the EffectParameter as a Vector4 .
 GetValueVector4Array	Gets the value of the EffectParameter as an array of Vector4 .
 ReferenceEquals	(Inherited from Object .)
 SetArrayRange	Sets the range of an array to pass to the device.
 SetValue	Overloaded. Sets the value of the EffectParameter .
 SetValueTranspose	Overloaded. Sets the value of the EffectParameter .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference



[EffectParameter Class](#)

EffectParameter Methods

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetValueBoolean	Gets the value of the EffectParameter as a Boolean.
 GetValueBooleanArray	Gets the value of the EffectParameter as an array of Boolean.
 GetValueInt32	Gets the value of the EffectParameter as an Int32.
 GetValueInt32Array	Gets the value of the EffectParameter as an array of Int32.
 GetValueMatrix	Gets the value of the EffectParameter as a Matrix .
 GetValueMatrixArray	Gets the value of the EffectParameter as an array of Matrix .
 GetValueMatrixTranspose	Gets the value of the EffectParameter as a Matrix transpose.
 GetValueMatrixTransposeArray	Gets the value of the EffectParameter as an array of Matrix transpose.
 GetValueQuaternion	Gets the value of the EffectParameter as a Quaternion .
 GetValueQuaternionArray	Gets the value of the EffectParameter as an array of Quaternion .
 GetValueSingle	Gets the value of the EffectParameter as a Single.
 GetValueSingleArray	Gets the value of the EffectParameter as an array of Single.
 GetValueString	Gets the value of the EffectParameter as a String.
 GetValueTexture2D	Gets the value of the EffectParameter as a Texture2D .
 GetValueTexture3D	Gets the value of the EffectParameter as a Texture3D .
 GetValueTextureCube	Gets the value of the EffectParameter as a TextureCube .
 GetValueVector2	Gets the value of the EffectParameter as a Vector2 .
 GetValueVector2Array	Gets the value of the EffectParameter as an array of Vector2 .
 GetValueVector3	Gets the value of the EffectParameter as a Vector3 .
 GetValueVector3Array	Gets the value of the EffectParameter as an array of Vector3 .
 GetValueVector4	Gets the value of the EffectParameter as a Vector4 .
 GetValueVector4Array	Gets the value of the EffectParameter as an array of Vector4 .
 ReferenceEquals	(Inherited from Object .)
 SetArrayRange	Sets the range of an array to pass to the device.
 SetValue	Overloaded. Sets the value of the EffectParameter .
 SetValueTranspose	Overloaded. Sets the value of the EffectParameter .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectParameter Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameter.GetValueBoolean Method

Gets the value of the [EffectParameter](#) as a [Boolean](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool GetValueBoolean ()
```

Return Value

The value of the [EffectParameter](#) as a [Boolean](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Boolean .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueBooleanArray Method

Gets the value of the [EffectParameter](#) as an array of [Boolean](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool[] GetValueBooleanArray (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Boolean](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	When attempting to call GetValueBooleanArray which returns an array, you must request that at least one member be returned.

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueInt32 Method

Gets the value of the [EffectParameter](#) as an [Int32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int GetValueInt32 ()
```

Return Value

Gets the value of the [EffectParameter](#) as an [Int32](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Int32 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueInt32Array Method

Gets the value of the [EffectParameter](#) as an array of [Int32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int[] GetValueInt32Array (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Int32](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	When attempting to call GetValueInt32Array which returns an array, you must request that at least one member be returned.

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueMatrix Method

Gets the value of the [EffectParameter](#) as a [Matrix](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix GetValueMatrix ()
```

Return Value

The value of the [EffectParameter](#) as a [Matrix](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Matrix .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueMatrixArray Method

Gets the value of the [EffectParameter](#) as an array of [Matrix](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix[] GetValueMatrixArray (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Matrix](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	When attempting to call GetValueMatrixArray which returns an array, you must request that at least one member be returned.
InvalidCastException	Unable to cast this EffectParameter to an array of type Matrix .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueMatrixTranspose Method

Gets the value of the [EffectParameter](#) as a [Matrix](#) transpose.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix GetValueMatrixTranspose ()
```

Return Value

The value of the [EffectParameter](#) as a [Matrix](#) transpose.

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Matrix .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueMatrixTransposeArray Method

Gets the value of the [EffectParameter](#) as an array of [Matrix](#) transpose.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix[] GetValueMatrixTransposeArray (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Matrix](#) transpose.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>count</i> is less than or equal to zero. <i>count</i> must be greater than our equal to one.
InvalidCastException	Unable to cast this EffectParameter to an array of type Matrix .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueQuaternion Method

Gets the value of the [EffectParameter](#) as a [Quaternion](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion GetValueQuaternion ()
```

Return Value

The value of the [EffectParameter](#) as a [Quaternion](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Quaternion .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueQuaternionArray Method

Gets the value of the [EffectParameter](#) as an array of [Quaternion](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion[] GetValueQuaternionArray (
    int count
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Quaternion](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>count</i> is less than or equal to zero. <i>count</i> must be greater than or equal to one.
InvalidCastException	Unable to cast this EffectParameter to an array of type Quaternion .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueSingle Method

Gets the value of the [EffectParameter](#) as a [Single](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GetValueSingle ()
```

Return Value

The value of the [EffectParameter](#) as a [Single](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Single .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueSingleArray Method

Gets the value of the [EffectParameter](#) as an array of [Single](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float[] GetValueSingleArray (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Single](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>count</i> is less than or equal to zero. <i>count</i> must be greater than or equal to one.
InvalidCastException	Unable to cast this EffectParameter to an array of type Single .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueString Method

Gets the value of the [EffectParameter](#) as an [String](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string GetValueString ()
```

Return Value

The value of the [EffectParameter](#) as an [String](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type String .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueTexture2D Method

Gets the value of the [EffectParameter](#) as a [Texture2D](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetValueTexture2D ()
```

Return Value

The value of the [EffectParameter](#) as a [Texture2D](#).

Exceptions

Exception type	Condition
InvalidOperationException	The resource is not of type ResourceType.Texture2D .
InvalidCastException	Unable to cast this EffectParameter to type Texture2D .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueTexture3D Method

Gets the value of the [EffectParameter](#) as a [Texture3D](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture3D GetValueTexture3D ()
```

Return Value

The value of the [EffectParameter](#) as a [Texture3D](#).

Exceptions

Exception type	Condition
InvalidOperationException	The resource is not of type ResourceType.Texture3D .
InvalidCastException	Unable to cast this EffectParameter to type Texture3D .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueTextureCube Method

Gets the value of the [EffectParameter](#) as a [TextureCube](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCube GetValueTextureCube ()
```

Return Value

The value of the [EffectParameter](#) as a [TextureCube](#).

Exceptions

Exception type	Condition
InvalidOperationException	The resource is not of type ResourceType.TextureCube .
InvalidCastException	Unable to cast this EffectParameter to type TextureCube .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueVector2 Method

Gets the value of the [EffectParameter](#) as a [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 GetValueVector2 ()
```

Return Value

The value of the [EffectParameter](#) as a [Vector2](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an object of type Vector2 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueVector2Array Method

Gets the value of the [EffectParameter](#) as an array of [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2[] GetValueVector2Array (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Vector2](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>count</i> is less than or equal to zero. <i>count</i> must be greater than or equal to one.
InvalidCastException	Unable to cast this EffectParameter to an array of type Vector2 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueVector3 Method

Gets the value of the [EffectParameter](#) as a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 GetValueVector3 ()
```

Return Value

The value of the [EffectParameter](#) as a [Vector3](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an object of type Vector3 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueVector3Array Method

Gets the value of the [EffectParameter](#) as an array of [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3[] GetValueVector3Array (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Vector3](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>count</i> is less than or equal to zero. <i>count</i> must be greater than or equal to one.
InvalidCastException	Unable to cast this EffectParameter to an array of type Vector3 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueVector4 Method

Gets the value of the [EffectParameter](#) as a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 GetValueVector4 ()
```

Return Value

The value of the [EffectParameter](#) as a [Vector4](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Vector4 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.GetValueVector4Array Method

Gets the value of the [EffectParameter](#) as an array of [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4[] GetValueVector4Array (  
    int count  
)
```

Parameters

count

The number of elements in the array.

Return Value

The value of the [EffectParameter](#) as an array of [Vector4](#).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>count</i> is less than or equal to zero. <i>count</i> must be greater than or equal to one.
InvalidCastException	Unable to cast this EffectParameter to an array of type Vector4 .

Remarks

Getting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetArrayRange Method

Sets the range of an array to pass to the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetArrayRange (  
    int start,  
    int end  
)
```

Parameters

start

The start index.

end

The stop index.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method

Sets the value of the [EffectParameter](#).

Overload List

Name	Description
EffectParameter.SetValue (Boolean)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Boolean[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (Int32)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Int32[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (Matrix)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Matrix[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (Quaternion)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Quaternion[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (Single)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Single[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (String)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Texture)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Vector2)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Vector2[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (Vector3)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Vector3[])	Sets the value of the EffectParameter .
EffectParameter.SetValue (Vector4)	Sets the value of the EffectParameter .
EffectParameter.SetValue (Vector4[])	Sets the value of the EffectParameter .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameter.SetValue Method (Boolean)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    bool value  
)
```

Parameters

value

[[MarshalAsAttribute\(U1\)](#)] The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Boolean .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Boolean[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    bool[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Boolean .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Int32)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    int value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Int32 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Int32[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    int[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Int32 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Matrix)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Matrix value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Matrix .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Matrix[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Matrix[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Matrix .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Quaternion)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Quaternion value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Quaternion .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Quaternion[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Quaternion[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Quaternion .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Single)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    float value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Single .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Single[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    float[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Single .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (String)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    string value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type String .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Texture)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Texture value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
ObjectDisposedException	SetValue was called after <i>value</i> was disposed.
InvalidCastException	Unable to cast this EffectParameter to type Texture .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Vector2)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Vector2 value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Vector2 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Vector2[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Vector2[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Vector2 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Vector3)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Vector3 value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Vector3 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Vector3[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Vector3[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Vector3 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Vector4)

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Vector4 value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to type Vector4 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValue Method (Vector4[])

Sets the value of the [EffectParameter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    Vector4[] value  
)
```

Parameters

value

The value to assign to the [EffectParameter](#).

Exceptions

Exception type	Condition
InvalidCastException	Unable to cast this EffectParameter to an array of type Vector4 .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValueTranspose Method

Sets the value of the [EffectParameter](#).

Overload List

Name	Description
EffectParameter.SetValueTranspose (Matrix)	Sets the value of the EffectParameter to the transpose of a Matrix .
EffectParameter.SetValueTranspose (Matrix[])	Sets the value of the EffectParameter to the transpose of a Matrix .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameter.SetValueTranspose Method (Matrix)

Sets the value of the [EffectParameter](#) to the transpose of a [Matrix](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValueTranspose (  
    Matrix value  
)
```

Parameters

value

The value.

Exceptions

Exception type	Condition
InvalidCastException	This EffectParameter is not a matrix of the correct size to contain the transpose of the matrix specified in <i>value</i> .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.SetValueTranspose Method (Matrix[])

Sets the value of the [EffectParameter](#) to the transpose of a [Matrix](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValueTranspose (  
    Matrix[] value  
)
```

Parameters

value

The value.

Exceptions

Exception type	Condition
InvalidCastException	This EffectParameter is not a matrix of the correct size to contain the transpose of the matrix specified in <i>value</i> .

Remarks

Setting the value of an effect parameter is a slow operation. Avoid high-frequency calls.

See Also

Reference

[EffectParameter Class](#)










[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter Properties

Public Properties

	Name	Description
	Annotations	Gets the collection of EffectAnnotation objects for this parameter.
	ColumnCount	Gets the number of columns in the parameter description.
	Elements	Gets the collection of effect parameters.
	Name	Gets the name of the parameter.
	ParameterClass	Gets the class of the parameter.
	ParameterType	Gets the type of the parameter.
	RowCount	Gets the number of rows in the parameter description.
	Semantic	Gets the semantic meaning, or usage, of the parameter.
	StructureMembers	Gets the collection of structure members.

See Also

Reference

[EffectParameter Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameter.Annotations Property

Gets the collection of [EffectAnnotation](#) objects for this parameter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectAnnotationCollection Annotations { get; }
```

Property Value

The collection of [EffectAnnotation](#) objects.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.ColumnCount Property

Gets the number of columns in the parameter description.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ColumnCount { get; }
```

Property Value

The number of columns in the parameter description.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.Elements Property

Gets the collection of effect parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterCollection Elements { get; }
```

Property Value

The collection of effect parameters.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.Name Property

Gets the name of the parameter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of the parameter.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.ParameterClass Property

Gets the class of the parameter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterClass ParameterClass { get; }
```

Property Value

The parameter class.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.ParameterType Property

Gets the type of the parameter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterType ParameterType { get; }
```

Property Value

The parameter type.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.RowCount Property

Gets the number of rows in the parameter description.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int RowCount { get; }
```

Property Value

The number of rows in the parameter description.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.Semantic Property

Gets the semantic meaning, or usage, of the parameter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Semantic { get; }
```

Property Value

The semantic meaning of the parameter.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameter.StructureMembers Property

Gets the collection of structure members.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterCollection StructureMembers { get; }
```

Property Value

The collection of structure members.

See Also

Reference

[EffectParameter Class](#)

[EffectParameter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock Class

Represents an [EffectParameter](#) state block.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class EffectParameterBlock : IDisposable
```

See Also

Reference

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista










EffectParameterBlock Members

The following tables list the members exposed by the EffectParameterBlock type.



Public Constructors

Name	Description
 EffectParameterBlock	Initializes a new instance of the EffectParameterBlock class.

Public Methods

Name	Description
 Apply	Applies the parameter changes in the EffectParameterBlock to the current effect state.
 Begin	Begins capturing parameter state changes from the parent Effect .
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 End	Stops capturing state changes from the parent Effect .
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectParameterBlock Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterBlock Constructor

Initializes a new instance of the [EffectParameterBlock](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterBlock (  
    Effect parent  
)
```

Parameters

parent

The [Effect](#) used to generate the [EffectParameterBlock](#).

See Also

Reference

[EffectParameterBlock Class](#)










[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock Methods

Public Methods

Name	Description
 Apply	Applies the parameter changes in the EffectParameterBlock to the current effect state.
 Begin	Begins capturing parameter state changes from the parent Effect .
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 End	Stops capturing state changes from the parent Effect .
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectParameterBlock Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterBlock.Apply Method

Applies the parameter changes in the [EffectParameterBlock](#) to the current effect state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Apply ()
```

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock.Begin Method

Begins capturing parameter state changes from the parent [Effect](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
EffectParameterBlock.Dispose ()	Immediately releases the unmanaged resources used by this object.
EffectParameterBlock.Dispose (Boolean)	Releases the unmanaged resources used by the EffectParameterBlock and optionally releases the managed resources.

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterBlock.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock.Dispose Method (Boolean)

Releases the unmanaged resources used by the [EffectParameterBlock](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [EffectParameterBlock](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock.End Method

Stops capturing state changes from the parent [Effect](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void End ()
```

Exceptions

Exception type	Condition
InvalidOperationException	End was called, but Begin has not yet been called. Begin must be called successfully before End can be called.

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterBlock.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[EffectParameterBlock Class](#)

[EffectParameterBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterClass Enumeration

Defines classes that can be used for effect parameters or shader constants.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum EffectParameterClass
```

Members

Member name	Description
MatrixColumns	Constant is a column major matrix.
MatrixRows	Constant is a row major matrix.
Object	Constant is either a texture, a shader, or a string.
Scalar	Constant is a scalar.
Struct	Constant is a structure.
Vector	Constant is a vector.

See Also

Reference

[EffectParameter.ParameterClass Property](#)

[ShaderConstant.ParameterClass Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterCollection Class

Manipulates a collection of [EffectParameter](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectParameterCollection : IEnumerable<EffectParameter>
```

See Also

Reference

[EffectParameterCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








EffectParameterCollection Members

The following tables list the members exposed by the EffectParameterCollection type.



Public Properties

	Name	Description
	Count	Gets the number of EffectParameter objects in this EffectParameterCollection .
	Item	Overloaded. Gets an EffectParameter object.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the EffectParameterCollection .
	GetHashCode	(Inherited from Object .)
	GetParameterBySemantic	Gets the handle of a top-level parameter or a structure member parameter by looking up its semantic with a case-insensitive search.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[EffectParameterCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the EffectParameterCollection .
	GetHashCode	(Inherited from Object .)
	GetParameterBySemantic	Gets the handle of a top-level parameter or a structure member parameter by looking up its semantic with a case-insensitive search.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectParameterCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterCollection.GetEnumerator Method

Returns an enumerator that can iterate through the [EffectParameterCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<EffectParameter> GetEnumerator ()
```

Return Value

The iterator.

See Also

Reference

[EffectParameterCollection Class](#)

[EffectParameterCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterCollection.GetParameterBySemantic Method

Gets the handle of a top-level parameter or a structure member parameter by looking up its semantic with a case-insensitive search.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameter GetParameterBySemantic (  
    string semantic  
)
```

Parameters

semantic

The name of the semantic to search for.

Return Value

The first parameter that matches the semantic, or **null** if no matching parameter was found.

Remarks

Getting an effect parameter by semantic is a slow operation. Avoid-high frequency calls.

See Also

Reference

[EffectParameterCollection Class](#)



[EffectParameterCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of EffectParameter objects in this EffectParameterCollection .
	Item	Overloaded. Gets an EffectParameter object.

See Also

Reference

[EffectParameterCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterCollection.Count Property

Gets the number of [EffectParameter](#) objects in this [EffectParameterCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of [EffectParameter](#) objects in this [EffectParameterCollection](#).

See Also

Reference

[EffectParameterCollection Class](#)

[EffectParameterCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterCollection.Item Property

Gets an [EffectParameter](#) object.

Overload List

Name	Description
EffectParameterCollection.Item (Int32)	Gets a specific EffectParameter object by using an index value.
EffectParameterCollection.Item (String)	Gets a specific EffectParameter by name.

See Also

Reference

[EffectParameterCollection Class](#)

[EffectParameterCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectParameterCollection.Item Property (Int32)

Gets a specific [EffectParameter](#) object by using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameter this [
    int index
] { get; }
```

Property Value

The [EffectParameter](#) object at index *index*.

See Also

Reference

[EffectParameterCollection Class](#)

[EffectParameterCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterCollection.Item Property (String)

Gets a specific [EffectParameter](#) by name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameter this [
    string name
] { get; }
```

Property Value

The [EffectParameter](#) object named *name*.

See Also

Reference

[EffectParameterCollection Class](#)

[EffectParameterCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectParameterType Enumeration

Defines types that can be used for effect parameters or shader constants.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum EffectParameterType
```

Members

Member name	Description
Bool	Parameter is a Boolean. Any nonzero value passed in will be mapped to 1 (TRUE) before being written into the constant table; otherwise, the value will be set to 0 in the constant table.
Int32	Parameter is an integer. Any floating-point values passed in will be rounded off (to zero decimal places) before being written into the constant table.
PixelShader	Parameter is a pixel shader.
Sampler	Parameter is a sampler.
Sampler1D	Parameter is a 1D sampler.
Sampler2D	Parameter is a 2D sampler.
Sampler3D	Parameter is a 3D sampler.
SamplerCube	Parameter is a cube sampler.
Single	Parameter is a floating-point number.
String	Parameter is a string.
Texture	Parameter is a texture.
Texture1D	Parameter is a 1D texture.
Texture2D	Parameter is a 2D texture.
Texture3D	Parameter is a 3D texture.
TextureCube	Parameter is a cube texture.
VertexShader	Parameter is a vertex shader.
Void	Parameter is a void pointer.

See Also

Reference

[EffectParameter.ParameterType Property](#)

[ShaderConstant.ParameterType Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPass Class

Represents an effect pass.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectPass
```

Remarks

An effect pass represents a single pass for a related effect. Effects can have a single pass or multiple passes. Use the [Passes](#) property to determine the number of passes for a related effect technique.

See Also

Reference

[EffectPass Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Tasks



[How To: Create and Apply Custom Effects](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








EffectPass Members

The following tables list the members exposed by the EffectPass type.



Public Properties

	Name	Description
	Annotations	Gets the set of EffectAnnotation objects for this EffectPass .
	Name	Gets the name of this pass.

Public Methods

	Name	Description
	Begin	Begins a pass within the active technique.
	End	Ends a pass within the active technique.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[EffectPass Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPass Methods

Public Methods

	Name	Description
	Begin	Begins a pass within the active technique.
	End	Ends a pass within the active technique.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectPass Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPass.Begin Method

Begins a pass within the active technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

Remarks

An application sets one active pass (within one active technique) in the effect system by calling [Begin](#). An application signals the end of the active pass by calling [End](#). [Begin](#) and [End](#) must occur in a matching pair, within a matching pair of [Effect.Begin](#) and [Effect.End](#) calls.

If the application changes any effect state using any of the [Effect](#) methods inside of a [Begin/End](#) matching pair, the application must call [CommitChanges](#) to set the update the device with the state changes. If no state changes occur within a [Begin](#) and [End](#) matching pair, it is not necessary to call [CommitChanges](#).

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

[How To: Use BasicEffect](#)

[How To: Create Custom Texture Effects](#)

Reference

[EffectPass Class](#)

[EffectPass Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPass.End Method

Ends a pass within the active technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void End ()
```

Exceptions

Exception type	Condition
InvalidOperationException	End was called, but Begin has not yet been called. Begin must be called successfully before End can be called.

Remarks

An application signals the end of rendering an active pass by calling [End](#). Each [End](#) must be part of a matching pair of [Begin](#) and [End](#) calls.

[Begin](#) and [End](#) must occur in a matching pair, within a matching pair of [Effect.Begin](#) and [Effect.End](#) calls.

If the application changes any effect state using any of the [Effect](#) methods inside of a [Begin/End](#) matching pair, the application must call [CommitChanges](#) to set the update the device with the state changes. If no state changes occur within a [Begin](#) and [End](#) matching pair, it is not necessary to call [CommitChanges](#).

See Also

Tasks

[How To: Create and Apply Custom Effects](#)

[How To: Use BasicEffect](#)

[How To: Create Custom Texture Effects](#)

Reference

[EffectPass Class](#)



[EffectPass Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPass Properties

Public Properties

	Name	Description
	Annotations	Gets the set of EffectAnnotation objects for this EffectPass .
	Name	Gets the name of this pass.

See Also

Reference

[EffectPass Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPass.Annotations Property

Gets the set of [EffectAnnotation](#) objects for this [EffectPass](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectAnnotationCollection Annotations { get; }
```

Property Value

The [EffectAnnotationCollection](#) containing [EffectAnnotation](#) objects for this [EffectPass](#).

See Also

Reference

[EffectPass Class](#)

[EffectPass Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPass.Name Property

Gets the name of this pass.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this pass.

See Also

Reference

[EffectPass Class](#)

[EffectPass Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPassCollection Class

Manipulates a collection of [EffectPass](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectPassCollection : IEnumerable<EffectPass>
```

See Also

Reference

[EffectPassCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista







EffectPassCollection Members

The following tables list the members exposed by the EffectPassCollection type.



Public Properties

	Name	Description
	Count	Gets the number of objects in the collection.
	Item	Overloaded. Gets an element in the collection.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the collection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[EffectPassCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPassCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the collection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectPassCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPassCollection.GetEnumerator Method

Returns an enumerator that can iterate through the collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<EffectPass> GetEnumerator ()
```

Return Value

Enumerator that can iterate through the collection.

See Also

Reference

[EffectPassCollection Class](#)



[EffectPassCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPassCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of objects in the collection.
	Item	Overloaded. Gets an element in the collection.

See Also

Reference

[EffectPassCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPassCollection.Count Property

Gets the number of objects in the collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of objects in the collection.

See Also

Reference

[EffectPassCollection Class](#)

[EffectPassCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPassCollection.Item Property

Gets an element in the collection.

Overload List

Name	Description
EffectPassCollection.Item (Int32)	Gets a specific element in the collection by using an index value.
EffectPassCollection.Item (String)	Gets a specific element in the collection by using a name.

See Also

Reference

[EffectPassCollection Class](#)

[EffectPassCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPassCollection.Item Property (Int32)

Gets a specific element in the collection by using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectPass this [
    int index
] { get; }
```

Property Value

The object at index *index*.

See Also

Reference

[EffectPassCollection Class](#)

[EffectPassCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPassCollection.Item Property (String)

Gets a specific element in the collection by using a name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectPass this [
    string name
] { get; }
```

Property Value

The object named *name*.

See Also

Reference

[EffectPassCollection Class](#)

[EffectPassCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool Class

Allows applications to share resources between effects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class EffectPool : IDisposable
```

Remarks

Effect pools allow different effects to share variables, textures, and shaders between them. This means that for common variables, the data needs to be set only once per frame, rather than once per effect invocation. An effect pool effectively consists of two parts. There is the effect pool itself, and the effects that are part of the pool.

A good example of sharing variables would be the projection matrix of the view. In many cases, the projection matrix is constant over an entire frame. However, every effect used in the frame needs that matrix at some point. If there are several hundred effects running per scene, you would have to set that matrix for each effect. However, by using effect pools, you set the projection matrix once, and that value is automatically used by every effect in the pool when needed.

Note

The content manager uses a static [EffectPool](#) when loading all effects. This supports the usage of shared parameters across all effects loaded by the content manager.

For each effect pool, parameters with the same name, type, and semantic are automatically shared across all effects in that pool. An easy way to implement this is an include file that declares each effect variable to be shared and then include that variable file in each effect file needing those variables.

For an example of effect pools, see Shader Series 1: Vertex Lighting, located at <http://creators.xna.com>.

See Also

Reference

[EffectPool Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


EffectPool Members

The following tables list the members exposed by the EffectPool type.







Public Constructors

Name	Description
 EffectPool	Initializes a new instance of this class.




Public Properties

Name	Description
 IsDisposed	Gets a value that indicates whether the object is disposed.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[EffectPool Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPool Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectPool ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Unable to create this EffectPool resource on the graphics device.

See Also

Reference

[EffectPool Class](#)







[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[EffectPool Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPool.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
EffectPool.Dispose ()	Immediately releases the unmanaged resources used by this object.
EffectPool.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

Remarks

Call [Dispose](#) when you are finished using the [EffectPool](#). The [Dispose](#) method leaves the [EffectPool](#) in an unusable state. After calling [Dispose](#), you must release all references to the [EffectPool](#) so the garbage collector can reclaim the memory that the [EffectPool](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [EffectPool](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [EffectPool](#) object's [Finalize](#) method.

See Also

Reference

[EffectPool Class](#)

[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPool.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [EffectPool](#). The [Dispose](#) method leaves the [EffectPool](#) in an unusable state. After calling [Dispose](#), you must release all references to the [EffectPool](#) so the garbage collector can reclaim the memory that the [EffectPool](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [EffectPool](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [EffectPool](#) object's [Finalize](#) method.

See Also

Reference

[EffectPool Class](#)

[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [EffectPool](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[EffectPool Class](#)

[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[EffectPool Class](#)

[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the Disposing event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[EffectPool Class](#)

[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

EffectPool.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[EffectPool Class](#)


[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.

See Also

Reference

[EffectPool Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPool.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[EffectPool Class](#)


[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectPool Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[EffectPool Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectPool.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

Remarks [IsDisposed](#) indicates whether that object has been disposed.

Example

To add an event handler that listens for the **Disposing** event, use the following C# code.

```
obj.Disposing += new System.EventHandler( this.OnDisposing );
```

See Also

Reference

[EffectPool Class](#)

[EffectPool Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechnique Class

Represents an effect technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectTechnique
```

Remarks Creating and assigning a **EffectTechnique** instance for each technique in your **Effect** is significantly faster than using the **Techniques** indexed property on **Effect**.

Example

To use a **EffectTechnique** you must:

1. Create a **EffectTechnique** for each technique in your **Effect**.

C#

```
public EffectTechnique texture;
public EffectTechnique shadows;
public EffectTechnique shadowMap;
```

2. Assign an **Effect** technique to your **EffectTechnique**.

C#

```
texture = effect.Techniques["TextureRender"];
shadowMap = effect.Techniques["ShadowMapRender"];
shadows = effect.Techniques["ShadowRender"];
```

3. Assign your **EffectTechnique** to the **CurrentTechnique** of your **Effect** before drawing.

C#

```
private void DrawScene(EffectTechnique technique)
{
    MyEffect.mWorld.SetValue(terrainWorld);
    MyEffect.MeshTexture.SetValue(terrainTex);
    foreach (ModelMesh mesh in terrain.Meshes)
    {
        foreach (Effect effect in mesh.Effects)
        {
            effect.CurrentTechnique = technique;
            mesh.Draw();
        }
    }
}
```

See Also

Concepts

[Shader Content Catalog at XNA Creators Club Online](#)

Tasks

[How To: Use EffectParameters and EffectTechniques](#)

Reference

[EffectTechnique Members](#)

[Effect.Techniques](#)




[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








EffectTechnique Members

The following tables list the members exposed by the EffectTechnique type.



Public Properties

	Name	Description
	Annotations	Gets the EffectAnnotation objects associated with this technique.
	Name	Gets the name of this technique.
	Passes	Gets the collection of EffectPass objects this rendering technique requires.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IsParameterUsed	Determines whether a given EffectParameter is used by this technique.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	Validates this technique.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[EffectTechnique Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechnique Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IsParameterUsed	Determines whether a given EffectParameter is used by this technique.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	Validates this technique.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectTechnique Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechnique.IsParameterUsed Method

Determines whether a given [EffectParameter](#) is used by this technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsParameterUsed (  
    EffectParameter parameter  
)
```

Parameters

parameter

The effect parameter to check.

Return Value

true if the parameter is used by this technique; **false** otherwise.

Exceptions

Exception type	Condition
ArgumentNullException	<i>parameter</i> is null .

See Also

Reference

[EffectTechnique Class](#)

[EffectTechnique Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechnique.Validate Method

Validates this technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Validate ()
```

Return Value

true if the technique is valid; **false** otherwise.

See Also

Reference

[EffectTechnique Class](#)




[EffectTechnique Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechnique Properties

Public Properties

	Name	Description
	Annotations	Gets the EffectAnnotation objects associated with this technique.
	Name	Gets the name of this technique.
	Passes	Gets the collection of EffectPass objects this rendering technique requires.

See Also

Reference

[EffectTechnique Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechnique.Annotations Property

Gets the [EffectAnnotation](#) objects associated with this technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectAnnotationCollection Annotations { get; }
```

Property Value

The [EffectAnnotation](#) objects associated with this technique.

See Also

Reference

[EffectTechnique Class](#)

[EffectTechnique Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechnique.Name Property

Gets the name of this technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this technique.

See Also

Reference

[EffectTechnique Class](#)

[EffectTechnique Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechnique.Passes Property

Gets the collection of [EffectPass](#) objects this rendering technique requires.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectPassCollection Passes { get; }
```

Property Value

The collection of [EffectPass](#) objects this rendering technique requires.

Remarks

Some video cards can render two textures in a single pass. However, if a card does not have this capability, it is often possible to render the same effect in two passes, using one texture for each pass.

See Also

Reference

[EffectTechnique Class](#)

[EffectTechnique Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechniqueCollection Class

Manipulates a collection of [EffectTechnique](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class EffectTechniqueCollection : IEnumerable<EffectTechnique>
```

See Also

Reference

[EffectTechniqueCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








EffectTechniqueCollection Members

The following tables list the members exposed by the EffectTechniqueCollection type.



Public Properties

	Name	Description
	Count	Gets the number of objects in the collection.
	Item	Overloaded. Gets an element in the collection.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the collection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetValidTechniques	Returns all of the valid techniques in this collection.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[EffectTechniqueCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechniqueCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the collection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetValidTechniques	Returns all of the valid techniques in this collection.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectTechniqueCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechniqueCollection.GetEnumerator Method

Returns an enumerator that can iterate through the collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<EffectTechnique> GetEnumerator ()
```

Return Value

Enumerator for the effect technique collection.

See Also

Reference

[EffectTechniqueCollection Class](#)

[EffectTechniqueCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechniqueCollection.GetValidTechniques Method

Returns all of the valid techniques in this collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerable<EffectTechnique> GetValidTechniques ()
```

Return Value

An enumeration of all the valid techniques in this collection.

See Also

Reference

[EffectTechniqueCollection Class](#)



[EffectTechniqueCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechniqueCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of objects in the collection.
	Item	Overloaded. Gets an element in the collection.

See Also

Reference

[EffectTechniqueCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechniqueCollection.Count Property

Gets the number of objects in the collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of objects in the collection.

See Also

Reference

[EffectTechniqueCollection Class](#)

[EffectTechniqueCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechniqueCollection.Item Property

Gets an element in the collection.

Overload List

Name	Description
EffectTechniqueCollection.Item (Int32)	Gets a specific element in the collection by using an index value.
EffectTechniqueCollection.Item (String)	Gets a specific element in the collection by using a name.

See Also

Reference

[EffectTechniqueCollection Class](#)

[EffectTechniqueCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

EffectTechniqueCollection.Item Property (Int32)

Gets a specific element in the collection by using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectTechnique this [
    int index
] { get; }
```

Property Value

The object at index *index*.

See Also

Reference

[EffectTechniqueCollection Class](#)

[EffectTechniqueCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

EffectTechniqueCollection.Item Property (String)

Gets a specific element in the collection by using a name.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectTechnique this [
    string name
] { get; }
```

Property Value

The object named *name*.

See Also

Reference

[EffectTechniqueCollection Class](#)

[EffectTechniqueCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

FillMode Enumeration

Describes options for filling the vertices and lines that define a primitive.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum FillMode
```

Members

Member name	Description
Point	Draw a point at each vertex.
Solid	Draw solid faces for each primitive.
WireFrame	Draw lines connecting the vertices that define a primitive face.

See Also

Reference

[RenderState.FillMode Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

FilterOptions Enumeration

Note

This enumeration is available only when developing for Windows.

Defines modes describing how to filter an image or mipmap when it is minified or magnified to fit a set of vertices.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum FilterOptions
```

Members

Member name	Description
Box	Each pixel is computed by averaging a 2×2(×2) box of pixels from the source image. This filter works only when the dimensions of the destination are half those of the source, as is the case with mipmaps.
Dither	Resulting image must be dithered using a 4×4 ordered dither algorithm. This happens when converting from one format to another.
DitherDiffusion	Do diffuse dithering on the image when changing from one format to another.
Linear	Bilinear interpolation filtering is used as a texture magnification or minification filter. A weighted average of a 2×2 area of texels surrounding the desired pixel is used. The texture filter to use between mipmap levels is trilinear mipmap interpolation. The rasterizer interpolates pixel color in a linear manner, using the texels of the two nearest textures.
Mirror	Same as specifying the MirrorU , MirrorV , and MirrorW flags. This flag is always used internally for this function.
MirrorU	Pixels off the edge of the texture on the u-axis should be mirrored, not wrapped.
MirrorV	Pixels off the edge of the texture on the v-axis should be mirrored, not wrapped.
MirrorW	Pixels off the edge of the texture on the w-axis should be mirrored, not wrapped.
None	Mipmapping disabled. The rasterizer uses the magnification filter instead.
Point	Each destination pixel is computed by sampling the nearest pixel from the source image.
Srgb	Same as specifying SrgbIn SrgbOut .
SrgbIn	Input data is in sRGB (gamma 2.2) color space.
SrgbOut	Output data is in sRGB (gamma 2.2) color space.
Triangle	Each pixel in the source image contributes equally to the destination image. This is the slowest of the filters.

Remarks Each valid filter must contain exactly one of the following flags: **None**, **Point**, **Linear**, **Triangle**, or **Box**. In addition, the bitwise OR operator can be used to specify zero or more of the following optional flags with a valid filter: **MirrorU**, **MirrorV**, **MirrorW**, **Mirror**, or **Dither**.

See Also

Reference

[TextureCreationParameters Constructor](#)

[TextureCreationParameters.Filter Property](#)

[TextureCreationParameters.MipFilter Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)
Platforms Windows XP SP2, Windows Vista

FogMode Enumeration

Note

This enumeration is available only when developing for Windows.

Defines constants that describe the fog mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum FogMode
```

Members

Member name	Description
Exponent	Fog effect intensifies exponentially, according to the following formula. $f = \frac{1}{e^{d \times density}}$
ExponentSquare	Fog effect intensifies exponentially with the square of the distance, according to the following formula. $f = \frac{1}{e^{(d \times density)^2}}$
Linear	Fog effect intensifies in a linear manner between the start and end points, according to the following formula. $f = \frac{end - d}{end - start}$
None	No fog effect.

Remarks

Fog can be considered a measure of visibility. The lower the fog value produced by a fog equation, the less visible an object is.

See Also

Reference

[RenderState.FogTableMode Property](#)

[RenderState.FogVertexMode Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GammaRamp Class

Contains red, green, and blue ramp data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GammaRamp : IDisposable
```

See Also

Reference

[GammaRamp Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista













GammaRamp Members

The following tables list the members exposed by the GammaRamp type.



Public Constructors

Name	Description
 GammaRamp	Initializes a new instance of this class.

Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetBlue	Retrieves the blue component of the gamma ramp.
 GetGreen	Retrieves the green component of the gamma ramp.
 GetHashCode	(Inherited from Object .)
 GetRed	Retrieves the red component of the gamma ramp.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetBlue	Sets the blue component of the gamma ramp.
 SetGreen	Sets the green component of the gamma ramp.
 SetRed	Sets the red component of the gamma ramp.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GammaRamp Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GammaRamp Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GammaRamp ()
```

See Also

Reference

[GammaRamp Class](#)













[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetBlue	Retrieves the blue component of the gamma ramp.
	GetGreen	Retrieves the green component of the gamma ramp.
	GetHashCode	(Inherited from Object .)
	GetRed	Retrieves the red component of the gamma ramp.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetBlue	Sets the blue component of the gamma ramp.
	SetGreen	Sets the green component of the gamma ramp.
	SetRed	Sets the red component of the gamma ramp.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GammaRamp Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GammaRamp.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
GammaRamp.Dispose ()	Immediately releases the unmanaged resources used by this object.
GammaRamp.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

Remarks

Call [Dispose](#) when you are finished using the [GammaRamp](#). The [Dispose](#) method leaves the [GammaRamp](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GammaRamp](#) so the garbage collector can reclaim the memory that the [GammaRamp](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GammaRamp](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GammaRamp](#) object's [Finalize](#) method.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GammaRamp.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [GammaRamp](#). The [Dispose](#) method leaves the [GammaRamp](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GammaRamp](#) so the garbage collector can reclaim the memory that the [GammaRamp](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GammaRamp](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GammaRamp](#) object's [Finalize](#) method.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [GammaRamp](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.GetBlue Method

Retrieves the blue component of the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public short[] GetBlue ()
```

Return Value

A 256-element integer array that represents the blue component.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.GetGreen Method

Retrieves the green component of the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public short[] GetGreen ()
```

Return Value

A 256-element integer array that represents the green component.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.GetRed Method

Retrieves the red component of the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public short[] GetRed ()
```

Return Value

A 256-element integer array that represents the red component.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.SetBlue Method

Sets the blue component of the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetBlue (  
    short[] value  
)
```

Parameters

value

A 256-element integer array that represents the blue component.

Exceptions

Exception type	Condition
ArgumentException	The value parameter is not a 256-element integer array. The arrays used in gamma ramps must have a length of 256.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.SetGreen Method

Sets the green component of the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetGreen (  
    short[] value  
)
```

Parameters

value

A 256-element integer array that represents the green component.

Exceptions

Exception type	Condition
ArgumentException	The value parameter is not a 256-element integer array. The arrays used in gamma ramps must have a length of 256.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GammaRamp.SetRed Method

Sets the red component of the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetRed (  
    short[] value  
)
```

Parameters

value

A 256-element integer array that represents the red component.

Exceptions

Exception type	Condition
ArgumentException	The value parameter is not a 256-element integer array. The arrays used in gamma ramps must have a length of 256.

See Also

Reference

[GammaRamp Class](#)

[GammaRamp Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsAdapter Class

Provides methods to retrieve and manipulate graphics adapters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class GraphicsAdapter : IDisposable
```

See Also

Reference

[GraphicsAdapter Members](#)

















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune














GraphicsAdapter Members


The following tables list the members exposed by the GraphicsAdapter type.

Public Properties



	Name	Description
	Adapters	Collection of available adapters on the system.
	CurrentDisplayMode	Gets the current display mode.
	DefaultAdapter	Gets the default adapter.
	Description	Retrieves a string used for presentation to the user.
	DeviceId	Retrieves a value that is used to help identify a particular chip set.
	DeviceIdentifier	Retrieves a globally unique identifier (GUID) object that can be used to check changes in the driver and chip set.
	DeviceName	Retrieves a string that contains the device name for a Microsoft Windows Graphics Device Interface (GDI).
	DriverDll	Retrieves a string that is used to present the driver name to the user.
	DriverVersion	Retrieves a value that identifies the version of the Microsoft Direct3D driver.
	IsDefaultAdapter	Determines if this instance of GraphicsAdapter is the default adapter.
	IsWideScreen	Determines if the graphics adapter is in widescreen mode.
	MonitorHandle	Retrieves the handle of the monitor associated with the Microsoft Direct3D object.
	Revision	Retrieves a value used to help identify the revision level of a particular chip set.
	SubSystemId	Retrieves a value used to identify the subsystem.
	SupportedDisplayModes	Returns a collection of supported display modes for the current adapter.
	VendorId	Retrieves a value used to identify the manufacturer.

Public Methods

	Name	Description
	CheckDepthStencilMatch	Tests whether a depth stencil format is compatible with a render target format in a particular display mode.
	CheckDeviceFormat	Overloaded. Determines whether a surface or depth buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three, on a device representing the current adapter.
	CheckDeviceFormatConversion	Tests a device to determine whether it supports conversion from one display format to another.
	CheckDeviceMultiSampleType	Overloaded. Verifies whether a multisample technique is available on a specified device.
	CheckDeviceType	Determines whether a hardware-accelerated device type can be used on the current adapter.
	Dispose	Overloaded. Releases all resources used by the GraphicsAdapter class.
	Equals	Overloaded. Determines whether two instances of GraphicsAdapter are equal.
	GetCapabilities	Returns the capabilities of a device.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	IsDeviceTypeAvailable	Determines whether a requested device type is available.
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.

 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also
















Reference

[GraphicsAdapter Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter Methods

Public Methods

	Name	Description
	CheckDepthStencilMatch	Tests whether a depth stencil format is compatible with a render target format in a particular display mode.
	CheckDeviceFormat	Overloaded. Determines whether a surface or depth buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three, on a device representing the current adapter.
	CheckDeviceFormatConversion	Tests a device to determine whether it supports conversion from one display format to another.
	CheckDeviceMultiSampleType	Overloaded. Verifies whether a multisample technique is available on a specified device.
	CheckDeviceType	Determines whether a hardware-accelerated device type can be used on the current adapter.
	Dispose	Overloaded. Releases all resources used by the GraphicsAdapter class.
	Equals	Overloaded. Determines whether two instances of GraphicsAdapter are equal.
	GetCapabilities	Returns the capabilities of a device.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	IsDeviceTypeAvailable	Determines whether a requested device type is available.
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsAdapter Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter.CheckDepthStencilMatch Method

Tests whether a depth stencil format is compatible with a render target format in a particular display mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDepthStencilMatch (
    DeviceType deviceType,
    SurfaceFormat adapterFormat,
    SurfaceFormat renderTargetFormat,
    DepthFormat depthStencilFormat
)
```

Parameters

deviceType

The device type.

adapterFormat

The format of the display mode into which the adapter will be placed.

renderTargetFormat

The format of the render-target surface to be tested.

depthStencilFormat

The format of the depth stencil surface to be tested.

Return Value

true if the format is compatible; **false** otherwise.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.
InvalidOperationException	An error has occurred.

Remarks

CheckDepthStencilMatch enables applications to work with hardware that requires that certain depth formats work only with certain render target formats.

Note

The following surface formats are not supported on the Xbox 360 platform for any [ResourceType](#), including textures:

- [SurfaceFormat.Bgr24](#)
- [SurfaceFormat.Bgra2338](#)
- [SurfaceFormat.Bgr233](#)
- [SurfaceFormat.NormalizedByte2Computed](#)
- [SurfaceFormat.LuminanceAlpha8](#)
- [SurfaceFormat.Palette8](#)
- [SurfaceFormat.PaletteAlpha16](#)
- [SurfaceFormat.Multi2Bgra32](#)
- [SurfaceFormat.Depth15Stencil1](#)
- [SurfaceFormat.Depth24Stencil4](#)

Example

C#

```
public static DepthStencilBuffer CreateDepthStencil(
    RenderTarget2D target)
{
    return new DepthStencilBuffer(target.GraphicsDevice, target.Width,
        target.Height, target.GraphicsDevice.DepthStencilBuffer.Format,
        target.MultiSampleType, target.MultiSampleQuality);
}
```

```
}  
public static DepthStencilBuffer CreateDepthStencil(  
    RenderTarget2D target, DepthFormat depth)  
{  
    if (GraphicsAdapter.DefaultAdapter.CheckDepthStencilMatch(  
        DeviceType.Hardware,  
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,  
        target.Format,  
        depth))  
    {  
        return new DepthStencilBuffer(target.GraphicsDevice,  
            target.Width, target.Height, depth,  
            target.MultiSampleType, target.MultiSampleQuality);  
    }  
    else  
        return CreateDepthStencil(target);  
}
```

See Also

Concepts

[Xbox 360 Surface Formats](#)

[What Is a Depth Buffer?](#)

[What Is a Stencil Buffer?](#)

[What Is a Render Target?](#)

Tasks

[How To: Create a Depth Texture](#)

Reference

[DepthStencilBuffer](#)

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CheckDeviceFormat Method

Determines whether a surface or depth buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three, on a device representing the current adapter.

Overload List

Name	Description
GraphicsAdapter.CheckDeviceFormat (DeviceType, SurfaceFormat, TextureUsage, QueryUsages, ResourceType, DepthFormat)	Verifies whether a depth buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three, on a device representing the current adapter.
GraphicsAdapter.CheckDeviceFormat (DeviceType, SurfaceFormat, TextureUsage, QueryUsages, ResourceType, SurfaceFormat)	Verifies whether a surface buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three on a device representing the current adapter.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter.CheckDeviceFormat Method (DeviceType, SurfaceFormat, TextureUsage, QueryUsages, ResourceType, DepthFormat)

Verifies whether a depth buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three, on a device representing the current adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDeviceFormat (
    DeviceType deviceType,
    SurfaceFormat adapterFormat,
    TextureUsage usage,
    QueryUsages queryUsages,
    ResourceType resourceType,
    DepthFormat checkFormat
)
```

Parameters

deviceType

The type of device to check.

adapterFormat

The format of the display mode into which the adapter will be placed.

usage

A set of options identifying the behaviors of this surface resource.

queryUsages

The type of query to issue.

resourceType

A resource type requested for use with the queried format.

checkFormat

The format of the depth buffer surface identified by *usage*.

Return Value

true if the surface format is available as the specified resource and format type; **false** otherwise.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.

Remarks

Note

The following surface formats are not supported on the Xbox 360 platform for any [ResourceType](#), including textures:

- [SurfaceFormat.Bgr24](#)
- [SurfaceFormat.Bgra2338](#)
- [SurfaceFormat.Bgr233](#)
- [SurfaceFormat.NormalizedByte2Computed](#)
- [SurfaceFormat.LuminanceAlpha8](#)
- [SurfaceFormat.Palette8](#)
- [SurfaceFormat.PaletteAlpha16](#)
- [SurfaceFormat.Multi2Bgra32](#)
- [SurfaceFormat.Depth15Stencil1](#)
- [SurfaceFormat.Depth24Stencil4](#)

Example

C#

```

public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}

```

See Also

Concepts

[Xbox 360 Surface Formats](#)

[How To: Create a Depth Texture](#)

Reference

[RenderTarget2D](#)

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CheckDeviceFormat Method (DeviceType, SurfaceFormat, TextureUsage, QueryUsages, ResourceType, SurfaceFormat)

Verifies whether a surface buffer format is available as a specified resource type and can be used as a texture, depth-stencil buffer, render target, or any combination of the three on a device representing the current adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDeviceFormat (
    DeviceType deviceType,
    SurfaceFormat adapterFormat,
    TextureUsage usage,
    QueryUsages queryUsages,
    ResourceType resourceType,
    SurfaceFormat checkFormat
)
```

Parameters

deviceType

The type of device to check.

adapterFormat

The format of the display mode into which the adapter will be placed.

usage

A set of options identifying the behaviors of this surface resource.

queryUsages

The type of query to issue.

resourceType

A resource type requested for use with the queried format.

checkFormat

The format of the surface identified by *usage*.

Return Value

true if the surface format is available as the specified resource and format type; **false** otherwise.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.
InvalidOperationException	An error has occurred.

Remarks

Note

The following surface formats are not supported on the Xbox 360 platform for any [ResourceType](#), including textures:

- [SurfaceFormat.Bgr24](#)
- [SurfaceFormat.Bgra2338](#)
- [SurfaceFormat.Bgr233](#)
- [SurfaceFormat.NormalizedByte2Computed](#)
- [SurfaceFormat.LuminanceAlpha8](#)
- [SurfaceFormat.Palette8](#)
- [SurfaceFormat.PaletteAlpha16](#)
- [SurfaceFormat.Multi2Bgra32](#)
- [SurfaceFormat.Depth15Stencil1](#)
- [SurfaceFormat.Depth24Stencil4](#)

Example

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

See Also

Concepts

[Xbox 360 Surface Formats](#)

[How To: Create a Depth Texture](#)

Reference

[RenderTarget2D](#)

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CheckDeviceFormatConversion Method

Tests a device to determine whether it supports conversion from one display format to another.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDeviceFormatConversion (
    DeviceType deviceType,
    SurfaceFormat sourceFormat,
    SurfaceFormat targetFormat
)
```

Parameters

deviceType

Type of device to check.

sourceFormat

Source adapter format.

targetFormat

Target adapter format.

Return Value

true if the method succeeds; **false** if it fails.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.
InvalidOperationException	An error has occurred.

Remarks

Using **CheckDeviceFormatConversion** to test for compatibility between a back buffer and the display format returns appropriate values. This means that the call reflects device capabilities. If the device cannot render to the requested back buffer format, the call still returns **false**. If the device can render to the format but cannot perform the color-converting presentation, the return value is also **false**. Applications can discover hardware support for the presentation itself by calling **CheckDeviceFormatConversion**. No software emulation for the color-converting presentation itself is offered.

Color conversion is restricted to certain source and target formats. The source format must be a four-character code (FOURCC) format or a valid back-buffer format, and the target format must be one of the following unsigned formats.

SurfaceFormat.Bgr555	SurfaceFormat.Bgra5551	SurfaceFormat.Bgr565
SurfaceFormat.Bgr24	SurfaceFormat.Bgr32	SurfaceFormat.Color
SurfaceFormat.Bgra1010102	SurfaceFormat.Rgba64	SurfaceFormat.Rgba1010102
SurfaceFormat.Rgba32	SurfaceFormat.Rgb32	SurfaceFormat.HalfVector4
SurfaceFormat.Vector4		

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CheckDeviceMultiSampleType Method

Verifies whether a multisample technique is available on a specified device.

Overload List

Name	Description
GraphicsAdapter.CheckDeviceMultiSampleType (DeviceType, SurfaceFormat, Boolean, MultiSampleType)	Verifies whether a multisample technique is available on a specified game machine.
GraphicsAdapter.CheckDeviceMultiSampleType (DeviceType, SurfaceFormat, Boolean, MultiSampleType, Int32)	Verifies whether a multisample technique is available on a specified game machine, optionally returning the number of quality stops available for the specified technique.

Remarks This method is intended for use with both render-target and depth-stencil surfaces because they must be created multisampled to be used together.

See Also

Concepts

[What Is Antialiasing?](#)

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter.CheckDeviceMultiSampleType Method (DeviceType, SurfaceFormat, Boolean, MultiSampleType)

Verifies whether a multisample technique is available on a specified game machine.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDeviceMultiSampleType (
    DeviceType deviceType,
    SurfaceFormat surfaceFormat,
    bool isFullScreen,
    MultiSampleType sampleType
)
```

Parameters

deviceType

The type of device to check.

surfaceFormat

The format of the surface to be multisampled.

isFullScreen

[[MarshalAsAttribute\(U1\)](#)] **true** to inquire about windowed multisampling; **false** to inquire about full-screen multisampling.

sampleType

The multisampling technique to test.

Return Value

true if the technique is available; **false** otherwise.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.

Example

Call **CheckDeviceMultiSampleType** in your [PreparingDeviceSettings](#) handler.

C#

```
int quality = 0;
GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
SurfaceFormat format = adapter.CurrentDisplayMode.Format;
// Check for 4xAA
if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
    false, MultiSampleType.FourSamples, out quality))
{
    // even if a greater quality is returned, we only want quality 0
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType =
        MultiSampleType.FourSamples;
}
// Check for 2xAA
else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
    format, false, MultiSampleType.TwoSamples, out quality))
{
    // even if a greater quality is returned, we only want quality 0
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType =
        MultiSampleType.TwoSamples;
}
return;
#endif
}
```

See Also

Concepts

[What Is Antialiasing?](#)

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CheckDeviceMultiSampleType Method (DeviceType, SurfaceFormat, Boolean, MultiSampleType, Int32)

Verifies whether a multisample technique is available on a specified game machine, optionally returning the number of quality stops available for the specified technique.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDeviceMultiSampleType (
    DeviceType deviceType,
    SurfaceFormat surfaceFormat,
    bool isFullScreen,
    MultiSampleType sampleType,
    out int qualityLevels
)
```

Parameters

deviceType

The type of device to check.

surfaceFormat

The format of the surface to be multisampled.

isFullScreen

[[MarshalAsAttribute\(U1\)](#)] **true** to inquire about windowed multisampling; **false** to inquire about full-screen multisampling.

sampleType

The multisampling technique to test.

qualityLevels

[[OutAttribute](#)] Number of quality stops available for a given multisample type; can be **null** if it is not necessary to return the values.

Return Value

true if the technique is available; **false** otherwise.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.
InvalidOperationException	An error has occurred.

Example

Call **CheckDeviceMultiSampleType** in your [PreparingDeviceSettings](#) handler.

C#

```
int quality = 0;
GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
SurfaceFormat format = adapter.CurrentDisplayMode.Format;
// Check for 4xAA
if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
    false, MultiSampleType.FourSamples, out quality))
{
    // even if a greater quality is returned, we only want quality 0
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType =
        MultiSampleType.FourSamples;
}
// Check for 2xAA
else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
    format, false, MultiSampleType.TwoSamples, out quality))
{
    // even if a greater quality is returned, we only want quality 0
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType =
```

```
        MultiSampleType.TwoSamples;
    }
    return;
#endif
}
```

See Also

Concepts

[What Is Antialiasing?](#)

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CheckDeviceType Method

Determines whether a hardware-accelerated device type can be used on the current adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CheckDeviceType (
    DeviceType deviceType,
    SurfaceFormat displayFormat,
    SurfaceFormat backBufferFormat,
    bool isFullScreen
)
```

Parameters

deviceType

The type of device to check.

displayFormat

The format of the adapter display mode for which the device type is being checked. For example, some devices operate only in modes of 16 bits per pixel.

backBufferFormat

Format of the back buffer. This value must be one of the render target formats. [DisplayMode](#) can be used to obtain the current format.

For windowed applications, the back-buffer format does not need to match the display-mode format if the hardware supports color conversion. The set of possible back-buffer formats is constrained, but the runtime allows any valid back-buffer format to be presented to any desktop format. Additionally, the device must be operable in desktop mode because devices typically do not operate in modes of 8 bits per pixel.

Full-screen applications cannot perform color conversion.

[SurfaceFormat.Unknown](#) is allowed for windowed mode.

isFullScreen

[[MarshalAsAttribute\(U1\)](#)] **true** if the device type will be used in windowed mode; **false** if the device type will be used in full-screen mode.

Return Value

true if the method succeeds and the device can be used on this adapter; **false** if the method fails. If the method fails, you may check the result code returned by [Manager.CheckDeviceTypeResult](#) to determine the specific reason.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.
InvalidOperationException	An error has occurred.

Remarks

A hardware abstraction layer (HAL) device type requires hardware acceleration. Applications can use **CheckDeviceType** to determine whether the hardware and drivers necessary to support a HAL device are present.

Full-screen applications should not specify a *displayFormat* that contains an alpha channel; doing so results in a failed call. Note that an alpha channel can be present in the back buffer, but the two display formats must be identical in all other respects. For example, if *displayFormat* = [SurfaceFormat.Bgr555](#), valid values for *backBufferFormat* include [SurfaceFormat.Bgr555](#) and [SurfaceFormat.Bgra5551](#), but exclude [SurfaceFormat.Bgr565](#).

Using **CheckDeviceType** to test for compatibility between a back buffer that differs from the display format returns appropriate values. This means that the call reflects device capabilities. No software emulation for the color-converting presentation itself is offered.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.Dispose Method

Releases all resources used by the [GraphicsAdapter](#) class.

Overload List

Name	Description
GraphicsAdapter.Dispose ()	Releases all resources used by the GraphicsAdapter class.
GraphicsAdapter.Dispose (Boolean)	Releases all resources used by the GraphicsAdapter class.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter.Dispose Method ()

Releases all resources used by the [GraphicsAdapter](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [GraphicsAdapter](#). The [Dispose](#) method leaves the [GraphicsAdapter](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GraphicsAdapter](#) so the garbage collector can reclaim the memory that the [GraphicsAdapter](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GraphicsAdapter](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GraphicsAdapter](#) object's [Finalize](#) method.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.Dispose Method (Boolean)

Note

This method is available only when developing for Windows.

Releases all resources used by the [GraphicsAdapter](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\] true](#) to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsAdapter.Equals Method

Determines whether two instances of [GraphicsAdapter](#) are equal.

Overload List

Name	Description
GraphicsAdapter.Equals (Object)	Determines whether the specified Object is equal to the GraphicsAdapter .
GraphicsAdapter.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter.Equals Method (Object)

Determines whether the specified [Object](#) is equal to the [GraphicsAdapter](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [GraphicsAdapter](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.GetCapabilities Method

Returns the capabilities of a device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDeviceCapabilities GetCapabilities (  
    DeviceType deviceType  
)
```

Parameters

deviceType

The device to query.

Return Value

The capabilities of the specified device.

Exceptions

Exception type	Condition
DeviceNotSupportedException	(Xbox 360) Hardware is the only device type supported on this platform.

See Also

Tasks

[How To: Check for Shader Model 2.0 Support](#)

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.IsDeviceTypeAvailable Method

Determines whether a requested device type is available.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDeviceTypeAvailable (  
    DeviceType deviceType  
)
```

Parameters

deviceType

The device type to request.

Return Value

true if the requested device type is available; **false** otherwise.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GraphicsAdapter left,  
    GraphicsAdapter right  
)
```

Parameters

left

The object to the left of the equality operator.

right

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GraphicsAdapter left,  
    GraphicsAdapter right  
)
```

Parameters

left

The object to the left of the inequality operator.

right

The object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[GraphicsAdapter Class](#)

















[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter Properties

Public Properties

	Name	Description
	Adapters	Collection of available adapters on the system.
	CurrentDisplayMode	Gets the current display mode.
	DefaultAdapter	Gets the default adapter.
	Description	Retrieves a string used for presentation to the user.
	DeviceId	Retrieves a value that is used to help identify a particular chip set.
	DeviceIdentifier	Retrieves a globally unique identifier (GUID) object that can be used to check changes in the driver and chip set.
	DeviceName	Retrieves a string that contains the device name for a Microsoft Windows Graphics Device Interface (GDI).
	DriverDll	Retrieves a string that is used to present the driver name to the user.
	DriverVersion	Retrieves a value that identifies the version of the Microsoft Direct3D driver.
	IsDefaultAdapter	Determines if this instance of GraphicsAdapter is the default adapter.
	IsWideScreen	Determines if the graphics adapter is in widescreen mode.
	MonitorHandle	Retrieves the handle of the monitor associated with the Microsoft Direct3D object.
	Revision	Retrieves a value used to help identify the revision level of a particular chip set.
	SubSystemId	Retrieves a value used to identify the subsystem.
	SupportedDisplayModes	Returns a collection of supported display modes for the current adapter.
	VendorId	Retrieves a value used to identify the manufacturer.

See Also

Reference

[GraphicsAdapter Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsAdapter.Adapters Property

Collection of available adapters on the system.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static ReadOnlyCollection<GraphicsAdapter> Adapters { get; }
```

Property Value

The collection of adapters.

See Also

Tasks

[How To: Check for Shader Model 2.0 Support](#)

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.CurrentDisplayMode Property

Gets the current display mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DisplayMode CurrentDisplayMode { get; }
```

Property Value

The current display mode.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.DefaultAdapter Property

Gets the default adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static GraphicsAdapter DefaultAdapter { get; }
```

Property Value

The default adapter.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.Description Property

Retrieves a string used for presentation to the user.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Description { get; }
```

Property Value

Description of the adapter.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.DeviceId Property

Retrieves a value that is used to help identify a particular chip set.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int DeviceId { get; }
```

Property Value

Value that helps identify the chip set. The returned value may be zero if it is unknown.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.DeviceIdentifier Property

Retrieves a globally unique identifier (GUID) object that can be used to check changes in the driver and chip set.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Guid DeviceIdentifier { get; }
```

Property Value

A unique identifier for the driver and chip set pair.

Remarks Use this member to track changes to the driver and chip set to generate a new profile for the graphics subsystem. It also can be used to identify problematic drivers.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.DeviceName Property

Retrieves a string that contains the device name for a Microsoft Windows Graphics Device Interface (GDI).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string DeviceName { get; }
```

Property Value

The name of the device.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.DriverDll Property

Retrieves a string that is used to present the driver name to the user.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string DriverDll { get; }
```

Property Value

The driver name

Remarks This member should not be used to identify particular drivers, because many different strings might be associated with the same device and driver from different vendors.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.DriverVersion Property

Retrieves a value that identifies the version of the Microsoft Direct3D driver.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Version DriverVersion { get; }
```

Property Value

The driver version.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.IsDefaultAdapter Property

Determines if this instance of [GraphicsAdapter](#) is the default adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDefaultAdapter { get; }
```

Property Value

true if this instance of [GraphicsAdapter](#) is the default adapter, **false** otherwise.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.IsWideScreen Property

Determines if the graphics adapter is in widescreen mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsWideScreen { get; }
```

Property Value

true if the graphics adapter is in widescreen mode; **false** otherwise.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.MonitorHandle Property

Retrieves the handle of the monitor associated with the Microsoft Direct3D object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IntPtr MonitorHandle { get; }
```

Property Value

The handle of the monitor.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.Revision Property

Retrieves a value used to help identify the revision level of a particular chip set.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Revision { get; }
```

Property Value

Value used to help identify the revision level of the chip set. This value may be zero if it is unknown.

Remarks The [VendorId](#), [DeviceId](#), [SubSystemId](#), and **Revision** members can be used in tandem to identify particular chip sets.

However, these members should be used with caution.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.SubSystemId Property

Retrieves a value used to identify the subsystem.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int SubSystemId { get; }
```

Property Value

Value used to identify the subsystem, typically the particular board. This value may be zero if it is unknown.

Remarks The [VendorId](#), [DeviceId](#), **SubSystemId**, and [Revision](#) members can be used in tandem to identify particular chip sets.

However, these members should be used with caution.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.SupportedDisplayModes Property

Returns a collection of supported display modes for the current adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DisplayModeCollection SupportedDisplayModes { get; }
```

Property Value

The supported display modes for the adapter.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsAdapter.VendorId Property

Retrieves a value used to identify the manufacturer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int VendorId { get; }
```

Property Value

Value used to identify the manufacturer. This value may be zero if it is unknown.

RemarksThe **VendorId**, [Deviceld](#), [SubSystemId](#), and [Revision](#) members can be used in tandem to identify particular chip sets.

However, these members should be used with caution.

See Also

Reference

[GraphicsAdapter Class](#)

[GraphicsAdapter Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice Class

Performs primitive-based rendering, creates resources, handles system-level variables, adjusts gamma ramp levels, and creates shaders.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GraphicsDevice : IDisposable
```

See Also

Reference

[GraphicsDevice Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






















GraphicsDevice Members

The following tables list the members exposed by the GraphicsDevice type.











Public Constructors

Name	Description
 GraphicsDevice	Creates a new instance of GraphicsDevice







Public Properties

Name	Description
 ClipPlanes	Retrieves the clipping planes of the current GraphicsDevice.
 CreationParameters	Retrieves the creation parameters of the GraphicsDevice.
 DepthStencilBuffer	Gets or sets the depth stencil surface of this GraphicsDevice .
 DisplayMode	Retrieves the display mode's spatial resolution, color resolution, and refresh frequency.
 DriverLevel	Returns the driver level.
 GraphicsDeviceCapabilities	Gets the capabilities of the graphics device.
 GraphicsDeviceStatus	Retrieves the status of the device
 Indices	Gets or sets index data.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 PixelShader	Gets or sets the current pixel shader.
 PresentationParameters	Gets the presentation parameters associated with this graphics device.
 RasterStatus	Retrieves information that describes the raster of the monitor on which the swap chain is presented.
 RenderState	Retrieves a render-state value for a GraphicsDevice.
 SamplerStates	Retrieves a collection of SamplerState objects for the current GraphicsDevice .
 ScissorRectangle	Gets or sets the rectangle used for scissor testing.
 Textures	Returns the collection of textures that have been assigned to the texture stages of the device.
 VertexDeclaration	Gets or sets a vertex shader declaration.
 VertexSamplerStates	Gets the collection of vertex sampler states.
 VertexShader	Gets or sets the current vertex shader.
 VertexTextures	Gets the collection of vertex textures that support texture lookup in the vertex shader using the texldl - vs texture load statement. The vertex engine contains four texture sampler stages.
 Vertices	Gets the vertex stream collection.
 Viewport	Gets or sets a viewport identifying the portion of the render target to receive draw calls.









Public Methods

Name	Description
 Clear	Overloaded. Clears the viewport depth buffer, and erases the stencil buffer.
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 DrawIndexedPrimitives	Renders the specified geometric primitive, based on indexing into an array of vertices.
 DrawPrimitives	Renders a sequence of non-indexed geometric primitives of the specified type from the current set of data input streams.
 DrawUserIndexedPrimitives	Overloaded. Renders the specified geometric primitive with data specified by the user.
 DrawUserPrimitives	Renders the given geometric primitive with data specified by the user.
 Equals	(Inherited from Object .)
 EvictManagedResources	Evicts all managed resources, including Microsoft Direct3D resources and those that are driver managed.
 GetGammaRamp	Gets the gamma correction ramp.
 GetHashCode	(Inherited from Object .)







GetPixelShaderBooleanConstant	Gets an array of Boolean values from the pixel shader constant Boolean registers.
GetPixelShaderInt32Constant	Gets an array of Int32 values from the pixel shader constant integer registers.
GetPixelShaderMatrixArrayConstant	Gets an array of Matrix values from the pixel shader constant float registers.
GetPixelShaderMatrixConstant	Gets a Matrix value from the pixel shader constant float registers.
GetPixelShaderQuaternionArrayConstant	Gets an array of Quaternion values from the pixel shader constant float registers.
GetPixelShaderQuaternionConstant	Gets a Quaternion value from the pixel shader constant float registers.
GetPixelShaderSingleConstant	Gets an array of Single values from the pixel shader constant float registers.
GetPixelShaderVector2ArrayConstant	Gets an array of Vector2 values from the pixel shader constant float registers.
GetPixelShaderVector2Constant	Gets a Vector2 value from the pixel shader constant float registers.
GetPixelShaderVector3ArrayConstant	Gets an array of Vector3 values from the pixel shader constant float registers.
GetPixelShaderVector3Constant	Gets a Vector3 value from the pixel shader constant float registers.
GetPixelShaderVector4ArrayConstant	Gets an array of Vector4 values from the pixel shader constant float registers.
GetPixelShaderVector4Constant	Gets a Vector4 value from the pixel shader constant float registers.
GetRenderTarget	Gets a render target surface.
GetType	(Inherited from Object .)
GetVertexShaderBooleanConstant	Gets an array of Boolean values from the vertex shader constant Boolean registers.
GetVertexShaderInt32Constant	Gets an array of Int32 values from the vertex shader constant integer registers.
GetVertexShaderMatrixArrayConstant	Gets an array of Matrix values from the vertex shader constant float registers.
GetVertexShaderMatrixConstant	Gets a Matrix value from the vertex shader constant float registers.
GetVertexShaderQuaternionArrayConstant	Gets an array of Quaternion values from the vertex shader constant float registers.
GetVertexShaderQuaternionConstant	Gets a Quaternion value from the vertex shader constant float registers.
GetVertexShaderSingleConstant	Gets an array of Single values from the vertex shader constant float registers.
GetVertexShaderVector2ArrayConstant	Gets an array of Vector2 values from the vertex shader constant float registers.
GetVertexShaderVector2Constant	Gets a Vector2 value from the vertex shader constant float registers.
GetVertexShaderVector3ArrayConstant	Gets an array of Vector3 values from the vertex shader constant float registers.
GetVertexShaderVector3Constant	Gets a Vector3 value from the vertex shader constant float registers.
GetVertexShaderVector4ArrayConstant	Gets an array of Vector4 values from the vertex shader constant float registers.
GetVertexShaderVector4Constant	Gets a Vector4 value from the vertex shader constant float registers.
Present	Overloaded. Presents the display with the contents of the next buffer in the sequence of back buffers owned by the GraphicsDevice .
ReferenceEquals	(Inherited from Object .)
Reset	Overloaded. Resets the presentation parameters for the current GraphicsDevice .

 ResolveBackBuffer	Overloaded. Copies the current back buffer contents to a texture.
 SetGammaRamp	Sets the gamma correction ramp.
 SetPixelShaderConstant	Overloaded. Sets a pixel shader constant.
 SetRenderTarget	Overloaded. Sets a new color buffer for a GraphicsDevice.
 SetVertexShaderConstant	Overloaded. Sets a vertex shader constant.
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_DeviceLost	Occurs when a GraphicsDevice is about to be lost (for example, immediately before a reset).
 raise_DeviceReset	Occurs after a GraphicsDevice is reset, allowing an application to re-create all resources.
 raise_DeviceResetting	Occurs when a GraphicsDevice is resetting
 raise_Disposing	Raises the Disposing event when called from within a derived class.
 raise_ResourceCreated	Occurs when ResourceCreated is called.
 raise_ResourceDestroyed	Occurs when ResourceDestroyed is called.

Public Events

Name	Description
 DeviceLost	Occurs when a GraphicsDevice is about to be lost (for example, immediately before a reset).
 DeviceReset	Occurs after a GraphicsDevice is reset, allowing an application to recreate all resources.
 DeviceResetting	Occurs when a GraphicsDevice is resetting, allowing the application to cancel the default handling of the reset.
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.
 ResourceCreated	Occurs when a resource is created.
 ResourceDestroyed	Occurs when a resource is destroyed.

See Also

Reference

[GraphicsDevice Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice Constructor

Creates a new instance of [GraphicsDevice](#)

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice (
    GraphicsAdapter adapter,
    DeviceType deviceType,
    IntPtr renderWindowHandle,
    PresentationParameters presentationParameters
)
```

Parameters

adapter

The display adapter.

deviceType

The desired device type.

renderWindowHandle

The focus window.

presentationParameters

The presentation parameters for the device to be created.

Exceptions

Exception type	Condition
ArgumentNullException	<i>adapter</i> or <i>presentationParameters</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> PresentationParameters.MultiSampleType contains an invalid value. The only valid MultiSampleType values on the Xbox 360 are MultiSampleType.None, MultiSampleType.TwoSamples, or MultiSampleType.FourSamples. For Windows-based applications, CheckDeviceMultiSampleType may be used to find valid values for this parameter. The back-buffer dimensions requested in PresentationParameters.BackBufferWidth and PresentationParameters.BackBufferHeight are too large for this graphics adapter.
InvalidOperationException	PresentationParameters.EnableAutoDepthStencil is true , but PresentationParameters.AutoDepthStencilFormat is not a valid value.
DeviceNotSupportedException	The graphics adapter does not support the requested capabilities. For Xbox 360, DeviceType.Hardware is the only device type supported on this platform.

Remarks

When you create a new XNA Framework game using XNA Game Studio, the game derives from [Game](#) and contains a member variable that is a [GraphicsDeviceManager](#). This [GraphicsDeviceManager](#) automatically creates the best device that is available for the application and exposes it as the [GraphicsDeviceManager.GraphicsDevice](#) property. If you wish to change the settings used to create the device, you can create a custom [PreparingDeviceSettings](#) event handler for the graphics device manager with the settings you require for your application. For more information on using the [PreparingDeviceSettings](#) event handler, see [GraphicsDeviceManager](#).

Note

If a graphics device has been constructed without the [GraphicsDeviceManager](#), it is necessary to manually call [Present](#) subsequent to any draw calls to present the back buffer.

See Also

Reference

[GraphicsDeviceManager Class](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









GraphicsDevice Methods

Public Methods

Name	Description
Clear	Overloaded. Clears the viewport depth buffer, and erases the stencil buffer.
Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
DrawIndexedPrimitives	Renders the specified geometric primitive, based on indexing into an array of vertices.
DrawPrimitives	Renders a sequence of non-indexed geometric primitives of the specified type from the current set of data input streams.
DrawUserIndexedPrimitives	Overloaded. Renders the specified geometric primitive with data specified by the user.
DrawUserPrimitives	Renders the given geometric primitive with data specified by the user.
Equals	(Inherited from Object .)
EvictManagedResources	Evicts all managed resources, including Microsoft Direct3D resources and those that are driver managed.
GetGammaRamp	Gets the gamma correction ramp.
GetHashCode	(Inherited from Object .)
GetPixelShaderBooleanConstant	Gets an array of Boolean values from the pixel shader constant Boolean registers.
GetPixelShaderInt32Constant	Gets an array of Int32 values from the pixel shader constant integer registers.
GetPixelShaderMatrixArrayConstant	Gets an array of Matrix values from the pixel shader constant float registers.
GetPixelShaderMatrixConstant	Gets a Matrix value from the pixel shader constant float registers.
GetPixelShaderQuaternionArrayConstant	Gets an array of Quaternion values from the pixel shader constant float registers.
GetPixelShaderQuaternionConstant	Gets a Quaternion value from the pixel shader constant float registers.
GetPixelShaderSingleConstant	Gets an array of Single values from the pixel shader constant float registers.
GetPixelShaderVector2ArrayConstant	Gets an array of Vector2 values from the pixel shader constant float registers.
GetPixelShaderVector2Constant	Gets a Vector2 value from the pixel shader constant float registers.
GetPixelShaderVector3ArrayConstant	Gets an array of Vector3 values from the pixel shader constant float registers.
GetPixelShaderVector3Constant	Gets a Vector3 value from the pixel shader constant float registers.
GetPixelShaderVector4ArrayConstant	Gets an array of Vector4 values from the pixel shader constant float registers.
GetPixelShaderVector4Constant	Gets a Vector4 value from the pixel shader constant float registers.
GetRenderTarget	Gets a render target surface.
GetType	(Inherited from Object .)
GetVertexShaderBooleanConstant	Gets an array of Boolean values from the vertex shader constant Boolean registers.
GetVertexShaderInt32Constant	Gets an array of Int32 values from the vertex shader constant integer registers.
GetVertexShaderMatrixArrayConstant	Gets an array of Matrix values from the vertex shader constant float registers.
GetVertexShaderMatrixConstant	Gets a Matrix value from the vertex shader constant float registers.
GetVertexShaderQuaternionArrayConstant	Gets an array of Quaternion values from the vertex shader constant float registers.
GetVertexShaderQuaternionConstant	Gets a Quaternion value from the vertex shader constant float registers.

 GetVertexShaderSingleConstant	Gets an array of Single values from the vertex shader constant float registers.
 GetVertexShaderVector2ArrayConstant	Gets an array of Vector2 values from the vertex shader constant float registers.
 GetVertexShaderVector2Constant	Gets a Vector2 value from the vertex shader constant float registers.
 GetVertexShaderVector3ArrayConstant	Gets an array of Vector3 values from the vertex shader constant float registers.
 GetVertexShaderVector3Constant	Gets a Vector3 value from the vertex shader constant float registers.
 GetVertexShaderVector4ArrayConstant	Gets an array of Vector4 values from the vertex shader constant float registers.
 GetVertexShaderVector4Constant	Gets a Vector4 value from the vertex shader constant float registers.
 Present	Overloaded. Presents the display with the contents of the next buffer in the sequence of back buffers owned by the GraphicsDevice .
 ReferenceEquals	(Inherited from Object .)
 Reset	Overloaded. Resets the presentation parameters for the current GraphicsDevice .
 ResolveBackBuffer	Overloaded. Copies the current back buffer contents to a texture.
 SetGammaRamp	Sets the gamma correction ramp.
 SetPixelShaderConstant	Overloaded. Sets a pixel shader constant.
 SetRenderTarget	Overloaded. Sets a new color buffer for a GraphicsDevice .
 SetVertexShaderConstant	Overloaded. Sets a vertex shader constant.
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_DeviceLost	Occurs when a GraphicsDevice is about to be lost (for example, immediately before a reset).
 raise_DeviceReset	Occurs after a GraphicsDevice is reset, allowing an application to re-create all resources.
 raise_DeviceResetting	Occurs when a GraphicsDevice is resetting
 raise_Disposing	Raises the Disposing event when called from within a derived class.
 raise_ResourceCreated	Occurs when ResourceCreated is called.
 raise_ResourceDestroyed	Occurs when ResourceDestroyed is called.

See Also

Reference

[GraphicsDevice Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.Clear Method

Clears the viewport depth buffer, and erases the stencil buffer.

Overload List

Name	Description
GraphicsDevice.Clear (ClearOptions, Color, Single, Int32)	Clears the viewport to a specified color, clears the depth buffer, and erases the stencil buffer.
GraphicsDevice.Clear (ClearOptions, Color, Single, Int32, Rectangle[])	Clears a set of regions to a specified color, clears the depth buffer, and erases the stencil buffer.
GraphicsDevice.Clear (ClearOptions, Vector4, Single, Int32)	Clears the viewport to a specified color, clears the depth buffer, and erases the stencil buffer.
GraphicsDevice.Clear (ClearOptions, Vector4, Single, Int32, Rectangle[])	Clears a set of regions to a specified color, clears the depth buffer, and erases the stencil buffer.
GraphicsDevice.Clear (Color)	Clears the viewport to a specified color.

Remarks This method fails if [ClearOptions.ZBuffer](#) or [ClearOptions.Stencil](#) is specified when the render target does not have an attached depth buffer. Similarly, the method fails if [ClearOptions.Stencil](#) is specified when the depth-buffer format does not contain stencil buffer information.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.Clear Method (ClearOptions, Color, Single, Int32)

Clears the viewport to a specified color, clears the depth buffer, and erases the stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear (
    ClearOptions options,
    Color color,
    float depth,
    int stencil
)
```

Parameters

options

Flags indicating which surfaces to clear.

color

Color value to which the render target surface is cleared.

depth

New depth value that this method stores in the depth buffer. This parameter can be in the range of 0.0 through 1.0 (for z-based or w-based depth buffers). A value of 0.0 represents the nearest distance to the viewer; a value of 1.0 represents the farthest distance.

stencil

Integer value to store in each stencil-buffer entry. This parameter can be in the range of 0 through $2n-1$, where n is the bit depth of the stencil buffer.

Exceptions

Exception type	Condition
InvalidOperationException	No DepthStencilBuffer surface exists, or the active render target and depth stencil surface do not have the same pixel size and multisampling type. Unable to clear the depth or stencil buffer.
ObjectDisposedException	Clear was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Clear Method (ClearOptions, Color, Single, Int32, Rectangle[])

Clears a set of regions to a specified color, clears the depth buffer, and erases the stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear (
    ClearOptions options,
    Color color,
    float depth,
    int stencil,
    Rectangle[] regions
)
```

Parameters

options

Flags that indicate which surfaces to clear.

color

Color value to which the render target surface is cleared.

depth

New depth value to store in the depth buffer. This parameter can be in the range of 0.0 through 1.0 (for z-based or w-based depth buffers). A value of 0.0 represents the nearest distance to the viewer; a value of 1.0 represents the farthest distance.

stencil

Integer value to store in each stencil-buffer entry. This parameter can be in the range of 0 through $2n-1$, where n is the bit depth of the stencil buffer.

regions

The regions to clear.

Exceptions

Exception type	Condition
InvalidOperationException	No DepthStencilBuffer surface exists, or the active render target and depth stencil surface do not have the same pixel size and multisampling type. Unable to clear the depth or stencil buffer.
ObjectDisposedException	Clear was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Clear Method (ClearOptions, Vector4, Single, Int32)

Clears the viewport to a specified color, clears the depth buffer, and erases the stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear (
    ClearOptions options,
    Vector4 color,
    float depth,
    int stencil
)
```

Parameters

options

The surfaces to clear.

color

Color value to which the render target surface is cleared.

depth

New z value that this method stores in the depth buffer. This parameter can be in the range of 0.0 through 1.0 (for z-based or w-based depth buffers). A value of 0.0 represents the nearest distance to the viewer; a value of 1.0 represents the farthest distance.

stencil

Integer value to store in each stencil-buffer entry. This parameter can be in the range of 0 through $2n-1$, where n is the bit depth of the stencil buffer.

Exceptions

Exception type	Condition
InvalidOperationException	No DepthStencilBuffer surface exists, or the active render target and depth stencil surface do not have the same pixel size and multisampling type. Unable to clear the depth or stencil buffer.
ObjectDisposedException	Clear was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Clear Method (ClearOptions, Vector4, Single, Int32, Rectangle[])

Clears a set of regions to a specified color, clears the depth buffer, and erases the stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear (
    ClearOptions options,
    Vector4 color,
    float depth,
    int stencil,
    Rectangle[] regions
)
```

Parameters

options

The surfaces to clear.

color

Color value to which the render target surface is cleared.

depth

New z value that this method stores in the depth buffer. This parameter can be in the range of 0.0 through 1.0 (for z-based or w-based depth buffers). A value of 0.0 represents the nearest distance to the viewer; a value of 1.0 represents the farthest distance.

stencil

Integer value to store in each stencil-buffer entry. This parameter can be in the range of 0 through $2n-1$, where n is the bit depth of the stencil buffer.

regions

The regions to clear.

Exceptions

Exception type	Condition
InvalidOperationException	No DepthStencilBuffer surface exists, or the active render target and depth stencil surface do not have the same pixel size and multisampling type. Unable to clear the depth or stencil buffer.
ObjectDisposedException	Clear was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Clear Method (Color)

Clears the viewport to a specified color.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear (  
    Color color  
)
```

Parameters

color

Color value to which the render target surface is cleared.

Exceptions

Exception type	Condition
ObjectDisposedException	Clear was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
GraphicsDevice.Dispose ()	Immediately releases the unmanaged resources used by this object.
GraphicsDevice.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

Remarks

Call [Dispose](#) when you are finished using the [GraphicsDevice](#). The [Dispose](#) method leaves the [GraphicsDevice](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GraphicsDevice](#) so the garbage collector can reclaim the memory that the [GraphicsDevice](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GraphicsDevice](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GraphicsDevice](#) object's [Finalize](#) method.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [GraphicsDevice](#). The [Dispose](#) method leaves the [GraphicsDevice](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GraphicsDevice](#) so the garbage collector can reclaim the memory that the [GraphicsDevice](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GraphicsDevice](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GraphicsDevice](#) object's [Finalize](#) method.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [GraphicsDevice](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.DrawIndexedPrimitives Method

Renders the specified geometric primitive, based on indexing into an array of vertices.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawIndexedPrimitives (
    PrimitiveType primitiveType,
    int baseVertex,
    int minVertexIndex,
    int numVertices,
    int startIndex,
    int primitiveCount
)
```

Parameters

primitiveType

Describes the type of primitive to render. [PrimitiveType.PointList](#) is not supported with this method.

baseVertex

Offset to add to each vertex index in the index buffer.

minVertexIndex

Minimum vertex index for vertices used during the call. The *minVertexIndex* parameter and all of the indices in the index stream are relative to the *baseVertex* parameter.

numVertices

A number of vertices used during the call. The first vertex is located at index: $baseVertex + minVertexIndex$.

startIndex

Location in the index array at which to start reading vertices.

primitiveCount

A number of primitives to render. The number of vertices used is a function of *primitiveCount* and *primitiveType*. To determine the maximum number of primitives allowed, check the **MaxPrimitiveCount** property member of the **Capabilities** structure.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>primitiveCount</i> is less than or equal to zero. When drawing, at least one primitive must be drawn.
InvalidOperationException	One of the following conditions is true: <ul style="list-style-type: none"> VertexDeclaration is not a valid value. A valid vertex declaration must be set on the device before any draw operations can be performed. A valid vertex shader and pixel shader was not set before calling DrawIndexedPrimitives. Both a valid vertex shader and pixel shader (or valid effect) must be set on the device before any draw operations may be performed. See How To: Use BasicEffect or How To: Create and Apply Custom Effects for more information on applying effects. Vertices is not a valid vertex buffer, or Indices is not a valid index buffer. A valid vertex buffer and a valid index buffer must be set on the device before any draw operations may be performed. The active render target and depth stencil surface do not have the same pixel size and multisampling type.

Example

The vertex stream and index data of the graphics device must be set before any call to [DrawIndexedPrimitives](#). The following example sets the index data and associates a user-created vertex buffer of type **VertexPositionNormalTexture** with vertex stream 0 (zero) of the graphics device.

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);  
  
graphics.GraphicsDevice.Indices = lineListIndexBuffer;
```

Remarks

This method draws indexed primitives from the current set of data input streams.

The *minVertexIndex* and *numVertices* parameters specify the range of vertex indices used for each call to **DrawIndexedPrimitives**. These vertex indices are used to optimize vertex processing of indexed primitives by processing a sequential range of vertices prior to indexing into them. Indices used during this call cannot reference any vertices outside this range.

If no index array is set, **DrawIndexedPrimitives** fails.

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.DrawPrimitives Method

Renders a sequence of non-indexed geometric primitives of the specified type from the current set of data input streams.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawPrimitives (
    PrimitiveType primitiveType,
    int startVertex,
    int primitiveCount
)
```

Parameters

primitiveType

Describes the type of primitive to render.

startVertex

Index of the first vertex to load. Beginning at *startVertex*, the correct number of vertices is read out of the vertex buffer.

primitiveCount

Number of primitives to render. To determine the maximum number of primitives allowed, check [MaxPrimitiveCount](#). The *primitiveCount* is the number of primitives as determined by the primitive type. If it is a line list, each primitive has two vertices. If it is a triangle list, each primitive has three vertices.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>primitiveCount</i> is less than or equal to zero. When drawing, at least one primitive must be drawn.
InvalidOperationException	One of the following conditions is true: <ul style="list-style-type: none"> VertexDeclaration is not a valid value. A valid vertex declaration must be set on the device before any draw operations can be performed. A valid vertex shader and pixel shader was not set before calling DrawPrimitives. Both a valid vertex shader and pixel shader (or valid effect) must be set on the device before any draw operations may be performed. See How To: Use BasicEffect or How To: Create and Apply Custom Effects for more information on applying effects. Vertices is not a valid vertex buffer. A valid vertex buffer must be set on the device before any draw operations may be performed. The active render target and depth stencil surface do not have the same pixel size and multisampling type.

Remarks

DrawPrimitives should not be called with a single triangle at a time.

Example

The vertex stream of the graphics device must be set before any call to [DrawPrimitives](#). The following example associates a user created vertex buffer of type **VertexPositionNormalTexture** with vertex stream 0 (zero) of the graphics device.

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(
    vertexBuffer, 0,
    VertexPositionNormalTexture.SizeInBytes);
```

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.DrawUserIndexedPrimitives Method

Renders the specified geometric primitive with data specified by the user.

Overload List

Name	Description
GraphicsDevice.DrawUserIndexedPrimitives (PrimitiveType, T[], Int32, Int32, Int16[], Int32, Int32)	Renders geometric primitives with indexed data specified by the user, specifying an index buffer as an array of type Int16 .
GraphicsDevice.DrawUserIndexedPrimitives (PrimitiveType, T[], Int32, Int32, Int32[], Int32, Int32)	Renders geometric primitives with indexed data specified by the user, specifying an index buffer as an array of type Int32 .

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.DrawUserIndexedPrimitives Generic Method (PrimitiveType, T[], Int32, Int32, Int16[], Int32, Int32)

Renders geometric primitives with indexed data specified by the user, specifying an index buffer as an array of type [Int16](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawUserIndexedPrimitives<T> (
    PrimitiveType primitiveType,
    T[] vertexData,
    int vertexOffset,
    int numVertices,
    short[] indexData,
    int indexOffset,
    int primitiveCount
) where T : ValueType
```

Type Parameters

T

The type of vertex in *vertexData*.

Parameters

primitiveType

Describes the type of primitive to render.

vertexData

The vertex buffer indexed by *indexData*.

vertexOffset

Offset to add to each vertex index in the index buffer.

numVertices

Number of vertices used during this call. The first vertex is located at index *minVertexIndex*.

indexData

A list of indices into the vertex buffer, given in the order that you want the vertices to render. Using an array of type [Int16](#), which uses 16 bits per element, allows you to conserve resources if the index buffer does not require a 32-bit depth array.

indexOffset

Location in the index array at which to start reading vertices.

primitiveCount

Number of primitives to render. The maximum number of primitives allowed is determined by checking [MaxPrimitiveCount](#). (The number of indices is a function of the primitive count and the primitive type.)

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> <i>vertexOffset</i> + <i>numVertices</i> is outside of the range of <i>vertexData</i>. There are not enough indices in <i>indexData</i> based on the number of primitives requested by <i>primitiveCount</i>.
ArgumentNullException	<i>vertexData</i> or <i>indexData</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> <i>indexOffset</i> or <i>vertexOffset</i> is not in a range allowed by <i>indexData</i> or <i>vertexData</i>. <i>primitiveCount</i> or <i>numVertices</i> is less than or equal to zero. When drawing, at least one primitive must be drawn.

InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none">• VertexDeclaration is not a valid value. A valid vertex declaration must be set on the device before any draw operations can be performed.• A valid vertex shader and pixel shader was not set before calling DrawUserIndexedPrimitives. Both a valid vertex shader and pixel shader (or valid effect) must be set on the device before any draw operations may be performed. See How To: Use BasicEffect or How To: Create and Apply Custom Effects for more information on applying effects.• <i>vertexData</i> is not a valid vertex buffer, or <i>indexData</i> is not a valid index buffer. A valid vertex buffer (and a valid index buffer if you are using indexed primitives) must be set on the device before any draw operations may be performed.• The active render target and depth stencil surface do not have the same pixel size and multisampling type.
---------------------------	---

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.DrawUserIndexedPrimitives Generic Method (PrimitiveType, T[], Int32, Int32, Int32[], Int32, Int32)

Renders geometric primitives with indexed data specified by the user, specifying an index buffer as an array of type [Int32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawUserIndexedPrimitives<T> (
    PrimitiveType primitiveType,
    T[] vertexData,
    int vertexOffset,
    int numVertices,
    int[] indexData,
    int indexOffset,
    int primitiveCount
) where T : ValueType
```

Type Parameters

T

The type of vertex in *vertexData*.

Parameters

primitiveType

Describes the type of primitive to render.

vertexData

The vertex buffer indexed by *indexData*.

vertexOffset

Offset to add to each vertex index in the index buffer.

numVertices

Number of vertices used during this call. The first vertex is located at index *minVertexIndex*.

indexData

A list of indices into the vertex buffer, given in the order that you want the vertices to render. Using an array of type [Int32](#), which uses 32 bits per element, allows you to index a greater number of elements in the vertex buffer.

indexOffset

Location in the index array at which to start reading vertices.

primitiveCount

Number of primitives to render. The maximum number of primitives allowed is determined by checking [MaxPrimitiveCount](#). (The number of indices is a function of the primitive count and the primitive type.)

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> <i>vertexOffset</i> + <i>numVertices</i> is outside of the range of <i>vertexData</i>. There are not enough indices in <i>indexData</i> based on the number of primitives requested by <i>primitiveCount</i>.
ArgumentNullException	<i>vertexData</i> or <i>indexData</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> <i>indexOffset</i> or <i>vertexOffset</i> is not in a range allowed by <i>indexData</i> or <i>vertexData</i>. <i>primitiveCount</i> or <i>numVertices</i> is less than or equal to zero. When drawing, at least one primitive must be drawn.

InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none">• VertexDeclaration is not a valid value. A valid vertex declaration must be set on the device before any draw operations can be performed.• A valid vertex shader and pixel shader was not set before calling DrawUserIndexedPrimitives. Both a valid vertex shader and pixel shader (or valid effect) must be set on the device before any draw operations may be performed. See How To: Use BasicEffect or How To: Create and Apply Custom Effects for more information on applying effects.• <i>vertexData</i> is not a valid vertex buffer, or <i>indexData</i> is not a valid index buffer. A valid vertex buffer (and a valid index buffer if you are using indexed primitives) must be set on the device before any draw operations may be performed.• The active render target and depth stencil surface do not have the same pixel size and multisampling type.
---------------------------	---

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.DrawUserPrimitives Generic Method

Renders the given geometric primitive with data specified by the user.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawUserPrimitives<T> (
    PrimitiveType primitiveType,
    T[] vertexData,
    int vertexOffset,
    int primitiveCount
) where T : ValueType
```

Type Parameters

T

Parameters

primitiveType

Describes the type of primitive to render.

vertexData

The vertex data.

vertexOffset

Offset at which to begin reading *vertexData*.

primitiveCount

Number of primitives to render. The maximum number of primitives allowed is determined by checking [MaxPrimitiveCount](#). The number of indices is a function of *primitiveCount* and *primitiveType*.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>vertexOffset</i> + <i>numVertices</i> is outside of the range of <i>vertexData</i> .
ArgumentNullException	<i>vertexData</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> <i>primitiveCount</i> is less than or equal to zero. When drawing, at least one primitive must be drawn. <i>vertexOffset</i> is less than zero or greater than the length of <i>vertexData</i>.
InvalidOperationException	One of the following conditions is true: <ul style="list-style-type: none"> VertexDeclaration is not a valid value. A valid vertex declaration must be set on the device before any draw operations can be performed. A valid vertex shader and pixel shader was not set before calling DrawUserPrimitives. Both a valid vertex shader and pixel shader (or valid effect) must be set on the device before any draw operations may be performed. See How To: Use BasicEffect or How To: Create and Apply Custom Effects for more information on applying effects. <i>vertexData</i> is not a valid vertex buffer. A valid vertex buffer must be set on the device before any draw operations may be performed.

Remarks

This method is intended for use when drawing dynamic geometry specified by the user. It supports only a single vertex stream. The effect of this call is to use the provided vertex data pointer and stride for vertex stream 0. It is invalid to have the declaration of the current vertex shader refer to vertex streams other than stream 0.

The vertex data passed to **DrawUserPrimitives** does not need to persist after the call. Access to the data is completed before

returning from the call.

See Also

Tasks

[How To: Draw Point Sprites](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.EvictManagedResources Method

Evicts all managed resources, including Microsoft Direct3D resources and those that are driver managed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void EvictManagedResources ()
```

Exceptions

Exception type	Condition
OutOfVideoMemoryException	Direct3D does not have enough display memory to perform the operation.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is invoked automatically during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.GetGammaRamp Method

Gets the gamma correction ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GammaRamp GetGammaRamp ()
```

Return Value

The gamma correction ramp.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderBooleanConstant Method

Gets an array of [Boolean](#) values from the pixel shader constant Boolean registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool[] GetPixelShaderBooleanConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader Boolean constant register of the first constant.

constantCount

Number of [Boolean](#) values to retrieve.

Return Value

Array of [Boolean](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderInt32Constant Method

Gets an array of [Int32](#) values from the pixel shader constant integer registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int[] GetPixelShaderInt32Constant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader integer register of the first constant.

constantCount

Number of [Int32](#) values to retrieve.

Return Value

Array of [Int32](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderMatrixArrayConstant Method

Gets an array of [Matrix](#) values from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix[] GetPixelShaderMatrixArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

constantCount

Number of [Matrix](#) values to retrieve.

Return Value

Array of [Matrix](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderMatrixConstant Method

Gets a [Matrix](#) value from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix GetPixelShaderMatrixConstant (  
    int startRegister  
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

Return Value

[Matrix](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderQuaternionArrayConstant Method

Gets an array of [Quaternion](#) values from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion[] GetPixelShaderQuaternionArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

constantCount

Number of [Quaternion](#) values to retrieve.

Return Value

Array of [Quaternion](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderQuaternionConstant Method

Gets a [Quaternion](#) value from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion GetPixelShaderQuaternionConstant (  
    int startRegister  
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

Return Value

[Quaternion](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderSingleConstant Method

Gets an array of [Single](#) values from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float[] GetPixelShaderSingleConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

constantCount

Number of [Single](#) values to retrieve.

Return Value

Array of [Single](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderVector2ArrayConstant Method

Gets an array of [Vector2](#) values from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2[] GetPixelShaderVector2ArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

constantCount

Number of [Vector2](#) values to retrieve.

Return Value

Array of [Vector2](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderVector2Constant Method

Gets a [Vector2](#) value from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 GetPixelShaderVector2Constant (
    int startRegister
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

Return Value

[Vector2](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderVector3ArrayConstant Method

Gets an array of [Vector3](#) values from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3[] GetPixelShaderVector3ArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

constantCount

Number of [Vector3](#) values to retrieve.

Return Value

Array of [Vector3](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderVector3Constant Method

Gets a [Vector3](#) value from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 GetPixelShaderVector3Constant (
    int startRegister
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

Return Value

[Vector3](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderVector4ArrayConstant Method

Gets an array of [Vector4](#) values from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4[] GetPixelShaderVector4ArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

constantCount

Number of [Vector4](#) values to retrieve.

Return Value

Array of [Vector4](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetPixelShaderVector4Constant Method

Gets a [Vector4](#) value from the pixel shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 GetPixelShaderVector4Constant (  
    int startRegister  
)
```

Parameters

startRegister

Pixel shader constant float register of the first constant.

Return Value

[Vector4](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetRenderTarget Method

Gets a render target surface.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTarget GetRenderTarget (  
    int renderTargetIndex  
)
```

Parameters

renderTargetIndex

Index of the render target.

Return Value

The render target surface of the current graphics device.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	An invalid render target index was requested. On Xbox 360, there is only one render target available and <i>renderTargetIndex</i> must be zero.

Remarks

The [GraphicsDevice](#) can support multiple render targets. The number of render targets supported by a GraphicsDevice is contained in [MaxSimultaneousRenderTargets](#).

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.GetVertexShaderBooleanConstant Method

Gets an array of [Boolean](#) values from the vertex shader constant Boolean registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool[] GetVertexShaderBooleanConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant Boolean register of the first constant.

constantCount

Number of [Boolean](#) values to retrieve.

Return Value

Array of [Boolean](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderInt32Constant Method

Gets an array of [Int32](#) values from the vertex shader constant integer registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int[] GetVertexShaderInt32Constant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant integer register of the first constant.

constantCount

Number of [Int32](#) values to retrieve.

Return Value

Array of [Int32](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderMatrixArrayConstant Method

Gets an array of [Matrix](#) values from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix[] GetVertexShaderMatrixArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

constantCount

Number of [Matrix](#) values to retrieve.

Return Value

Array of [Matrix](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderMatrixConstant Method

Gets a [Matrix](#) value from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix GetVertexShaderMatrixConstant (  
    int startRegister  
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

Return Value

[Matrix](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderQuaternionArrayConstant Method

Gets an array of [Quaternion](#) values from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion[] GetVertexShaderQuaternionArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

constantCount

Number of [Quaternion](#) values to retrieve.

Return Value

Array of [Quaternion](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderQuaternionConstant Method

Gets a [Quaternion](#) value from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion GetVertexShaderQuaternionConstant (  
    int startRegister  
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

Return Value

[Quaternion](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderSingleConstant Method

Gets an array of [Single](#) values from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float[] GetVertexShaderSingleConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

constantCount

Number of [Single](#) values to retrieve.

Return Value

Array of [Single](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderVector2ArrayConstant Method

Gets an array of [Vector2](#) values from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2[] GetVertexShaderVector2ArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

constantCount

Number of [Vector2](#) values to retrieve.

Return Value

Array of [Vector2](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderVector2Constant Method

Gets a [Vector2](#) value from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 GetVertexShaderVector2Constant (  
    int startRegister  
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

Return Value

[Vector2](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderVector3ArrayConstant Method

Gets an array of [Vector3](#) values from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3[] GetVertexShaderVector3ArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

constantCount

Number of [Vector3](#) values to retrieve.

Return Value

Array of [Vector3](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderVector3Constant Method

Gets a [Vector3](#) value from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 GetVertexShaderVector3Constant (  
    int startRegister  
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

Return Value

[Vector3](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderVector4ArrayConstant Method

Gets an array of [Vector4](#) values from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4[] GetVertexShaderVector4ArrayConstant (
    int startRegister,
    int constantCount
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

constantCount

Number of [Vector4](#) values to retrieve.

Return Value

Array of [Vector4](#) values retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GetVertexShaderVector4Constant Method

Gets a [Vector4](#) value from the vertex shader constant float registers.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 GetVertexShaderVector4Constant (  
    int startRegister  
)
```

Parameters

startRegister

Vertex shader constant float register of the first constant.

Return Value

[Vector4](#) value retrieved from the constant registers.

Exceptions

Exception type	Condition
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.Present Method

Presents the display with the contents of the next buffer in the sequence of back buffers owned by the [GraphicsDevice](#).

Overload List

Name	Description
GraphicsDevice.Present ()	Presents the display with the contents of the next buffer in the sequence of back buffers owned by the GraphicsDevice .
GraphicsDevice.Present (IntPtr)	Specifies the window target for a presentation and presents the display with the contents of the next buffer in the sequence of back buffers owned by the GraphicsDevice .
GraphicsDevice.Present (Nullable<Rectangle>, Nullable<Rectangle>, IntPtr)	Specifies the window target for a presentation and presents the display with the contents of the next buffer in the sequence of back buffers owned by the GraphicsDevice .

Exceptions

Exception type	Condition
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.

Remarks

If necessary, a stretch operation is applied to transfer the pixels within the source rectangle to the destination rectangle in the client area of the target window.

See Also

Reference

[Reset](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.Present Method ()

Presents the display with the contents of the next buffer in the sequence of back buffers owned by the [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Present ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	Present was called after the GraphicsDevice was disposed.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewport.Width,
    leftViewport.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);

// TODO: Add your code to draw to the render target here.
// For example, this could be a call to Mesh.Draw,
// a SpriteBatch Begin-End sequence, or a call to
// DrawIndexedPrimitives, as you would have called them
// in the Draw method of your application.

// Set the device render target back to the back buffer.
graphics.GraphicsDevice.SetRenderTarget(0, null);

// Call GetTexture to retrieve the render target data and save it to a texture.
capturedTexture = renderTarget.GetTexture();
```

See Also

Reference

[Present](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Present Method (IntPtr)

Specifies the window target for a presentation and presents the display with the contents of the next buffer in the sequence of back buffers owned by the [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Present (
    IntPtr overrideWindowHandle
)
```

Parameters

overrideWindowHandle

Destination window containing the client area that is the target for this presentation. If not specified, this is [DeviceWindowHandle](#).

Exceptions

Exception type	Condition
ObjectDisposedException	Present was called after the GraphicsDevice was disposed.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewPort.Width,
    leftViewPort.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);

// TODO: Add your code to draw to the render target here.
// For example, this could be a call to Mesh.Draw,
// a SpriteBatch Begin-End sequence, or a call to
// DrawIndexedPrimitives, as you would have called them
// in the Draw method of your application.

// Set the device render target back to the back buffer.
graphics.GraphicsDevice.SetRenderTarget(0, null);

// Call GetTexture to retrieve the render target data and save it to a texture.
capturedTexture = renderTarget.GetTexture();
```

See Also

Reference

[Present](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Present Method (Nullable<Rectangle>, Nullable<Rectangle>, IntPtr)

Specifies the window target for a presentation and presents the display with the contents of the next buffer in the sequence of back buffers owned by the [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Present (
    Nullable<Rectangle> sourceRectangle,
    Nullable<Rectangle> destinationRectangle,
    IntPtr overrideWindowHandle
)
```

Parameters

sourceRectangle

The source rectangle. If **null**, the entire source surface is presented. If the rectangle exceeds the source surface, the rectangle is clipped to the source surface. This parameter must be **null** unless the swap chain was created with [SwapEffect.Copy](#).

destinationRectangle

The destination rectangle, in window client coordinates. If **null**, the entire client area is filled. If the rectangle exceeds the destination client area, the rectangle is clipped to the destination client area. This parameter must be **null** unless the swap chain was created with [SwapEffect.Copy](#).

overrideWindowHandle

Destination window containing the client area that is the target for this presentation. If not specified, this is [DeviceWindowHandle](#).

Exceptions

Exception type	Condition
ObjectDisposedException	Present was called after the GraphicsDevice was disposed.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewPort.Width,
    leftViewPort.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);

// TODO: Add your code to draw to the render target here.
// For example, this could be a call to Mesh.Draw,
// a SpriteBatch Begin-End sequence, or a call to
// DrawIndexedPrimitives, as you would have called them
// in the Draw method of your application.

// Set the device render target back to the back buffer.
graphics.GraphicsDevice.SetRenderTarget(0, null);

// Call GetTexture to retrieve the render target data and save it to a texture.
```

```
capturedTexture = renderTarget.GetTexture();
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.raise_DeviceLost Method

Note

This method is available only when developing for Windows.

Occurs when a [GraphicsDevice](#) is about to be lost (for example, immediately before a reset).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_DeviceLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDevice.raise_DeviceReset Method

Note

This method is available only when developing for Windows.

Occurs after a [GraphicsDevice](#) is reset, allowing an application to re-create all resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_DeviceReset (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDevice.raise_DeviceResetting Method

Note

This method is available only when developing for Windows.

Occurs when a [GraphicsDevice](#) is resetting

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_DeviceResetting (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDevice.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDevice.raise_ResourceCreated Method

Note

This method is available only when developing for Windows.

Occurs when [ResourceCreated](#) is called.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_ResourceCreated (  
    Object value0,  
    ResourceCreatedEventArgs value1  
)
```

Parameters

value0

The source of the event.

value1

An **EventArgs** object that contains no data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDevice.raise_ResourceDestroyed Method

Note

This method is available only when developing for Windows.

Occurs when [ResourceDestroyed](#) is called.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_ResourceDestroyed (  
    Object value0,  
    ResourceDestroyedEventArgs value1  
)
```

Parameters

value0

The source of the event.

value1

An **EventArgs** object that contains no data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDevice.Reset Method

Resets the presentation parameters for the current GraphicsDevice.

Overload List

Name	Description
GraphicsDevice.Reset ()	Resets the presentation parameters for the current GraphicsDevice .
GraphicsDevice.Reset (GraphicsAdapter)	Resets the presentation parameters for the specified Reset .
GraphicsDevice.Reset (PresentationParameters)	Resets the current GraphicsDevice with the specified PresentationParameters .
GraphicsDevice.Reset (PresentationParameters, GraphicsAdapter)	Resets the specified Reset with the specified presentation parameters.

Exceptions

Exception type	Condition
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
OutOfVideoMemoryException	Direct3D does not have enough display memory to perform the operation.
OutOfMemoryException	Direct3D could not allocate sufficient memory to complete the call.

Remarks

When switching to full screen mode, Microsoft Direct3D tries to find a desktop format that matches the back buffer format, so that back and front buffer formats are identical. This eliminates the need for color conversion.

Calling **Reset** causes all texture memory surfaces and state information to be lost, and managed textures to be flushed from video memory. Before calling **Reset** for a [GraphicsDevice](#), an application should release any explicit render targets, depth stencil surfaces, additional swap chains, state blocks, and default resources associated with the [GraphicsDevice](#).

Swap chains are either full screen or windowed. If the new swap chain is full screen, the adapter is placed in the display mode that matches the new size.

A call to **Reset** fails if made on a different thread than the one used to create the [GraphicsDevice](#) being reset.

Pixel shaders and vertex shaders survive **Reset** calls for DirectX 9.0. They do not need to be re-created explicitly by the application.

Unknown can be specified for the windowed-mode back-buffer format when creating a [GraphicsDevice](#) and **Reset**. This means the application does not have to query the current desktop format before calling a [GraphicsDevice](#) constructor for windowed mode. For full screen mode, the back-buffer format must be specified. Setting [BackBufferCount](#) to 0 results in the creation of one back buffer.

When trying to reset more than one display adapter in a group, pass in an array of [PresentationParameters](#) objects, one for each display in the adapter group.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.Reset Method ()

Resets the presentation parameters for the current [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Reset ()
```

Exceptions

Exception type	Condition
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
OutOfVideoMemoryException	Direct3D does not have enough display memory to perform the operation.
OutOfMemoryException	Direct3D could not allocate sufficient memory to complete the call."

See Also

Reference

[Reset](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Reset Method (GraphicsAdapter)

Note

This method is available only when developing for Windows.

Resets the presentation parameters for the specified **Reset**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Reset (  
    GraphicsAdapter graphicsAdapter  
)
```

Parameters

graphicsAdapter

The graphics device being reset.

Exceptions

Exception type	Condition
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
OutOfVideoMemoryException	Direct3D does not have enough display memory to perform the operation.
OutOfMemoryException	Direct3D could not allocate sufficient memory to complete the call."

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDevice.Reset Method (PresentationParameters)

Resets the current [GraphicsDevice](#) with the specified [PresentationParameters](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Reset (  
    PresentationParameters presentationParameters  
)
```

Parameters

presentationParameters

Describes the new presentation parameters. This value cannot be **null**.

Exceptions

Exception type	Condition
ArgumentNullException	<i>presentationParameters</i> is null .
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
OutOfVideoMemoryException	Direct3D does not have enough display memory to perform the operation.
OutOfMemoryException	Direct3D could not allocate sufficient memory to complete the call.

See Also

Reference

[Reset](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Reset Method (PresentationParameters, GraphicsAdapter)

Resets the specified **Reset** with the specified presentation parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Reset (
    PresentationParameters presentationParameters,
    GraphicsAdapter graphicsAdapter
)
```

Parameters

presentationParameters

Describes the new presentation parameters. This value cannot be **null**.

graphicsAdapter

The graphics device being reset.

Exceptions

Exception type	Condition
ArgumentNullException	<i>presentationParameters</i> is null .
DeviceLostException	The GraphicsDevice is lost but cannot be reset at this time. Therefore, rendering is not possible.
DriverInternalErrorException	Internal driver error. Applications should generally shut down when receiving this error.
OutOfVideoMemoryException	Direct3D does not have enough display memory to perform the operation.
OutOfMemoryException	Direct3D could not allocate sufficient memory to complete the call."

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.ResolveBackBuffer Method

Copies the current back buffer contents to a texture.

Overload List

Name	Description
GraphicsDevice.ResolveBackBuffer (ResolveTexture2D)	Copies the current back buffer contents to a texture.
GraphicsDevice.ResolveBackBuffer (ResolveTexture2D, Int32)	Copies the contents of the back buffer at the specified index to a texture.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.ResolveBackBuffer Method (ResolveTexture2D)

Copies the current back buffer contents to a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ResolveBackBuffer (
    ResolveTexture2D resolveTarget
)
```

Parameters

resolveTarget

Texture to update with the resolved back buffer.

Exceptions

Exception type	Condition
InvalidOperationException	ResolveBackBuffer may not be called when the current render target is null . Use ResolveBackBuffer instead.

Remarks

On Xbox 360, the back buffer is cleared when **ResolveBackBuffer** is called.

Example

In this example, a new [Texture2D](#) is created for use as a render target, the data is initialized with a call to [ResolveBackBuffer](#), and the generation of mipmaps for the resolved render target is requested.

C#

```
ResolveTexture2D renderTargetTexture;
renderTargetTexture = new ResolveTexture2D(
    graphics.GraphicsDevice,
    graphics.GraphicsDevice.PresentationParameters.BackBufferWidth,
    graphics.GraphicsDevice.PresentationParameters.BackBufferHeight,
    1,
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

graphics.GraphicsDevice.ResolveBackBuffer(renderTargetTexture);
renderTargetTexture.GenerateMipMaps( TextureFilter.Linear );
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.ResolveBackBuffer Method (ResolveTexture2D, Int32)

Copies the contents of the back buffer at the specified index to a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ResolveBackBuffer (
    ResolveTexture2D resolveTarget,
    int backBufferIndex
)
```

Parameters

resolveTarget

Texture to update with the resolved back buffer.

backBufferIndex

Index of the back buffer to resolve.

Exceptions

Exception type	Condition
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> <i>backBufferIndex</i> is not valid. Note that on Xbox 360, there is only one back buffer available and <i>backBufferIndex</i> must be zero.
ArgumentNullException	<i>resolveTarget</i> is null
InvalidOperationException	ResolveBackBuffer may not be called when the current render target is null . Use ResolveBackBuffer instead.

Remarks

On Xbox 360, the back buffer is cleared when **ResolveBackBuffer** is called.

Example

In this example, a new [Texture2D](#) is created for use as a render target, the data is initialized with a call to [ResolveBackBuffer](#), and the generation of mipmaps for the resolved render target is requested.

C#

```
ResolveTexture2D renderTargetTexture;
renderTargetTexture = new ResolveTexture2D(
    graphics.GraphicsDevice,
    graphics.GraphicsDevice.PresentationParameters.BackBufferWidth,
    graphics.GraphicsDevice.PresentationParameters.BackBufferHeight,
    1,
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

graphics.GraphicsDevice.ResolveBackBuffer(renderTargetTexture);
renderTargetTexture.GenerateMipMaps( TextureFilter.Linear );
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.SetGammaRamp Method

Sets the gamma correction ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetGammaRamp (
    bool calibrate,
    GammaRamp ramp
)
```

Parameters

calibrate

[[MarshalAsAttribute\(U1\)](#)] **true** to indicate that correction should be applied. **false** to indicate that no gamma correction should be applied. The supplied gamma table is transferred directly to the [GraphicsDevice](#).

ramp

The gamma correction ramp to set.

Exceptions

Exception type	Condition
ArgumentNullException	<i>ramp</i> is null .
InvalidOperationException	The arrays used in the red, green, and blue components of <i>ramp</i> do not have an array length of 256. Arrays used for gamma ramps must have a length of 256.

Remarks

Gamma correction results in a more consistent display, but can incur processing overhead and should not be used frequently. Short-duration effects, such as flashing the entire screen red, should not be calibrated, but long-duration gamma changes should be. If a gamma calibrator is installed, the ramp is modified before being sent to the [GraphicsDevice](#) to account for the system and monitor response curves. If no calibrator is installed, the ramp is passed directly to the [GraphicsDevice](#).

The gamma ramp takes effect immediately. No wait for a vertical sync operation is performed.

If the [GraphicsDevice](#) does not support gamma ramps in the current presentation mode (full-screen or windowed), no error is returned. Applications can check [SupportsFullScreenGamma](#) and [CanCalibrateGamma](#) to determine the capabilities of the [GraphicsDevice](#) and whether a calibrator is installed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method

Sets a pixel shader constant.

Overload List

Name	Description
GraphicsDevice.SetPixelShaderConstant (Int32, Boolean[])	Sets the specified pixel shader constant Boolean registers to an array of Boolean values.
GraphicsDevice.SetPixelShaderConstant (Int32, Int32[])	Sets the specified pixel shader constant integer registers to an array of Int32 values.
GraphicsDevice.SetPixelShaderConstant (Int32, Matrix)	Sets the specified pixel shader constant float registers to a Matrix value.
GraphicsDevice.SetPixelShaderConstant (Int32, Matrix[])	Sets the specified pixel shader constant float registers to an array of Matrix values.
GraphicsDevice.SetPixelShaderConstant (Int32, Quaternion)	Sets the specified pixel shader constant float registers to a Quaternion value.
GraphicsDevice.SetPixelShaderConstant (Int32, Quaternion[])	Sets the specified pixel shader constant float registers to an array of Quaternion values.
GraphicsDevice.SetPixelShaderConstant (Int32, Single[])	Sets the specified pixel shader constant float registers to an array of Single values.
GraphicsDevice.SetPixelShaderConstant (Int32, Vector2)	Sets the specified pixel shader constant float register to a Vector2 value.
GraphicsDevice.SetPixelShaderConstant (Int32, Vector2[])	Sets the specified pixel shader constant float registers to an array of Vector2 values.
GraphicsDevice.SetPixelShaderConstant (Int32, Vector3)	Sets the specified pixel shader constant float register to a Vector3 value.
GraphicsDevice.SetPixelShaderConstant (Int32, Vector3[])	Sets the specified pixel shader constant float registers to an array of Vector3 values.
GraphicsDevice.SetPixelShaderConstant (Int32, Vector4)	Sets the specified pixel shader constant float register to a Vector4 value.
GraphicsDevice.SetPixelShaderConstant (Int32, Vector4[])	Sets the specified pixel shader constant float registers to an array of Vector4 values.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.SetPixelShaderConstant Method (Int32, Boolean[])

Sets the specified pixel shader constant Boolean registers to an array of [Boolean](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    bool[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant Boolean register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Int32[])

Sets the specified pixel shader constant integer registers to an array of [Int32](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    int[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant integer register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentException	When setting integer shader constant data, <i>constantData</i> must have a length that is a multiple of four.
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Matrix)

Sets the specified pixel shader constant float registers to a [Matrix](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Matrix constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Matrix[])

Sets the specified pixel shader constant float registers to an array of [Matrix](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Matrix[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Quaternion)

Sets the specified pixel shader constant float registers to a [Quaternion](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Quaternion constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Quaternion[])

Sets the specified pixel shader constant float registers to an array of [Quaternion](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Quaternion[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Single[])

Sets the specified pixel shader constant float registers to an array of [Single](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    float[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentException	When setting float shader constant data, <i>constantData</i> must have a length that is a multiple of four.
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Vector2)

Sets the specified pixel shader constant float register to a [Vector2](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Vector2 constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector2](#) type is not large enough to fill the constant float register, the constant data will be padded to fill the register. This means that setting register 0 to a [Vector2](#) containing (1,2) will fill the register with (1,2,0,0).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>constantData</i> is outside the allowable range of values as defined by the invoked method.
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Vector2[])

Sets the specified pixel shader constant float registers to an array of [Vector2](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Vector2[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector2](#) type is not large enough to fill the constant float registers, the constant data will be padded to fill the registers. This means that setting register 0 to an array of [Vector2](#) values containing (1,2) and (3,4) will fill register 0 with (1,2,0,0) and register 1 with (3,4,0,0).

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Vector3)

Sets the specified pixel shader constant float register to a [Vector3](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Vector3 constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

RemarksBecause the data in the [Vector3](#) type is not large enough to fill the constant float register, the constant data will be padded to fill the register. This means that setting register 0 to a [Vector3](#) containing (1,2,3) will fill the register with (1,2,3,0).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Vector3[])

Sets the specified pixel shader constant float registers to an array of [Vector3](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Vector3[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector3](#) type is not large enough to fill the constant float registers, the constant data will be padded to fill the registers. This means that setting register 0 to an array of [Vector3](#) values containing (1,2,3) and (4,5,6) will fill register 0 with (1,2,3,0) and register 1 with (4,5,6,0).

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Vector4)

Sets the specified pixel shader constant float register to a [Vector4](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Vector4 constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetPixelShaderConstant Method (Int32, Vector4[])

Sets the specified pixel shader constant float registers to an array of [Vector4](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetPixelShaderConstant (
    int startRegister,
    Vector4[] constantData
)
```

Parameters

startRegister

Zero-based index of the pixel shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetRenderTarget Method

Sets a new color buffer for a GraphicsDevice.

Overload List

Name	Description
GraphicsDevice.SetRenderTarget (Int32, RenderTarget2D)	Sets a new render target for this GraphicsDevice .
GraphicsDevice.SetRenderTarget (Int32, RenderTargetCube, CubeMapFace)	Sets a new render target for this GraphicsDevice .

See Also

Concepts

[What Is a Render Target?](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.SetRenderTarget Method (Int32, RenderTarget2D)

Sets a new render target for this **GraphicsDevice**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetRenderTarget (
    int renderTargetIndex,
    RenderTarget2D renderTarget
)
```

Parameters

renderTargetIndex

Index of the render target. The number of render targets supported by a [GraphicsDevice](#) is contained in [MaxSimultaneousRenderTargets](#).

renderTarget

A new render target for the device, or **null** to set the device render target to the back buffer of the device.

Exceptions

Exception type	Condition
ArgumentException	An invalid render target index was requested. On Xbox 360, there is only one render target available and <i>renderTargetIndex</i> must be zero.
ArgumentOutOfRangeException	The render target index must be within the valid range for this device.
ObjectDisposedException	SetRenderTarget was called after this <i>renderTarget</i> was disposed.

Remarks

The following restrictions apply when using this method.

- The multisample type must be the same for the render target and the depth stencil surface.
- The formats must be compatible for the render target and the depth stencil surface. For more information, see [CheckDepthStencilMatch](#).
- The size of the depth stencil surface must be greater than or equal to the size of the render target.

These restrictions are validated only when using the debug runtime when any of the [GraphicsDevice](#) drawing methods are called.

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewPort.Width,
    leftViewPort.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);

// TODO: Add your code to draw to the render target here.
// For example, this could be a call to Mesh.Draw,
```

```
// a SpriteBatch Begin-End sequence, or a call to
// DrawIndexedPrimitives, as you would have called them
// in the Draw method of your application.

// Set the device render target back to the back buffer.
graphics.GraphicsDevice.SetRenderTarget(0, null);

// Call GetTexture to retrieve the render target data and save it to a texture.
capturedTexture = renderTarget.GetTexture();
```

See Also

Concepts

[What Is a Render Target?](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.SetRenderTarget Method (Int32, RenderTargetCube, CubeMapFace)

Sets a new render target for this **GraphicsDevice**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetRenderTarget (
    int renderTargetIndex,
    RenderTargetCube renderTarget,
    CubeMapFace faceType
)
```

Parameters

renderTargetIndex

Index of the render target. The number of render targets supported by a [GraphicsDevice](#) is contained in [MaxSimultaneousRenderTargets](#).

renderTarget

A new render target for the device, or **null** to set the device render target to the back buffer of the device.

faceType

The cube map face type.

Exceptions

Exception type	Condition
ArgumentException	An invalid render target index was requested. On Xbox 360, there is only one render target available and <i>renderTargetIndex</i> must be zero.
ArgumentOutOfRangeException	The render target index must be within the valid range for this device.
ObjectDisposedException	SetRenderTarget was called after this <i>renderTarget</i> was disposed.

Remarks

The following restrictions apply when using this method.

- The multisample type must be the same for the render target and the depth stencil surface.
- The formats must be compatible for the render target and the depth stencil surface. For more information, see [CheckDepthStencilMatch](#).
- The size of the depth stencil surface must be greater than or equal to the size of the render target.

These restrictions are validated only when using the debug runtime when any of the [GraphicsDevice](#) drawing methods are called.

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewPort.Width,
    leftViewPort.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);
```



```
// TODO: Add your code to draw to the render target here.  
// For example, this could be a call to Mesh.Draw,  
// a SpriteBatch Begin-End sequence, or a call to  
// DrawIndexedPrimitives, as you would have called them  
// in the Draw method of your application.  
  
// Set the device render target back to the back buffer.  
graphics.GraphicsDevice.SetRenderTarget(0, null);  
  
// Call GetTexture to retrieve the render target data and save it to a texture.  
capturedTexture = renderTarget.GetTexture();
```

See Also

Concepts

[What Is a Render Target?](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method

Sets a vertex shader constant.

Overload List

Name	Description
GraphicsDevice.SetVertexShaderConstant (Int32, Boolean[])	Sets the specified vertex shader constant Boolean registers to an array of Boolean values.
GraphicsDevice.SetVertexShaderConstant (Int32, Int32[])	Sets the specified vertex shader constant integer registers to an array of Int32 values.
GraphicsDevice.SetVertexShaderConstant (Int32, Matrix)	Sets the specified vertex shader constant float registers to a Matrix value.
GraphicsDevice.SetVertexShaderConstant (Int32, Matrix[])	Sets the specified vertex shader constant float registers to an array of Matrix values.
GraphicsDevice.SetVertexShaderConstant (Int32, Quaternion)	Sets the specified vertex shader constant float register to a Quaternion value.
GraphicsDevice.SetVertexShaderConstant (Int32, Quaternion[])	Sets the specified vertex shader constant float registers to an array of Quaternion values.
GraphicsDevice.SetVertexShaderConstant (Int32, Single[])	Sets the specified vertex shader constant float registers to an array of Single values.
GraphicsDevice.SetVertexShaderConstant (Int32, Vector2)	Sets the specified vertex shader constant float register to a Vector2 value.
GraphicsDevice.SetVertexShaderConstant (Int32, Vector2[])	Sets the specified vertex shader constant float registers to an array of Vector2 values.
GraphicsDevice.SetVertexShaderConstant (Int32, Vector3)	Sets the specified vertex shader constant float register to a Vector3 value.
GraphicsDevice.SetVertexShaderConstant (Int32, Vector3[])	Sets the specified vertex shader constant float registers to an array of Vector3 values.
GraphicsDevice.SetVertexShaderConstant (Int32, Vector4)	Sets the specified vertex shader constant float register to a Vector4 value.
GraphicsDevice.SetVertexShaderConstant (Int32, Vector4[])	Sets the specified vertex shader constant float registers to an array of Vector4 values.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.SetVertexShaderConstant Method (Int32, Boolean[])

Sets the specified vertex shader constant Boolean registers to an array of [Boolean](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    bool[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant Boolean register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Int32[])

Sets the specified vertex shader constant integer registers to an array of [Int32](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    int[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant integer register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentException	When setting integer shader constant data, <i>constantData</i> must have a length that is a multiple of four.
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Matrix)

Sets the specified vertex shader constant float registers to a [Matrix](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Matrix constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data. This value will be transposed when the vertex shader constant is set.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Matrix[])

Sets the specified vertex shader constant float registers to an array of [Matrix](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Matrix[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data. This value will be transposed when the vertex shader constant is set.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Quaternion)

Sets the specified vertex shader constant float register to a [Quaternion](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Quaternion constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Quaternion[])

Sets the specified vertex shader constant float registers to an array of [Quaternion](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Quaternion[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Single[])

Sets the specified vertex shader constant float registers to an array of [Single](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    float[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentException	When setting float shader constant data, <i>constantData</i> must have a length that is a multiple of four.
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Vector2)

Sets the specified vertex shader constant float register to a [Vector2](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Vector2 constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector2](#) type is not large enough to fill the constant float register, the constant data will be padded to fill the register. This means that setting register 0 to a [Vector2](#) of of (1,2) will fill the register with (1,2,0,0).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>constantData</i> is outside the allowable range of values as defined by the invoked method.
InvalidOperationException	The method call is invalid. For example, a method's parameter may have an invalid value.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Vector2[])

Sets the specified vertex shader constant float registers to an array of [Vector2](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Vector2[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector2](#) type is not large enough to fill the constant float registers, the constant data will be padded to fill the registers. This means that setting register 0 to an array of [Vector2](#) values containing (1,2) and (3,4) will fill register 0 with (1,2,0,0) and register 1 with (3,4,0,0).

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Vector3)

Sets the specified vertex shader constant float register to a [Vector3](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Vector3 constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector3](#) type is not large enough to fill the constant float register, the constant data will be padded to fill the register. This means that setting register 0 to a [Vector3](#) of (1,2,3) will fill the register with (1,2,3,0).

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Vector3[])

Sets the specified vertex shader constant float registers to an array of [Vector3](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Vector3[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Remarks Because the data in the [Vector3](#) type is not large enough to fill the constant float registers, the constant data will be padded to fill the registers. This means that setting register 0 to to an array of [Vector3](#) values containing (1,2,3) and (4,5,6) will fill register 0 with (1,2,3,0) and register 1 with (4,5,6,0).

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Vector4)

Sets the specified vertex shader constant float register to a [Vector4](#) value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Vector4 constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SetVertexShaderConstant Method (Int32, Vector4[])

Sets the specified vertex shader constant float registers to an array of [Vector4](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetVertexShaderConstant (
    int startRegister,
    Vector4[] constantData
)
```

Parameters

startRegister

Zero-based index of the vertex shader constant float register at which to begin setting values.

constantData

The constant data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>constantData</i> is null .
ArgumentOutOfRangeException	The graphics buffer is not large enough for this data.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDevice Class](#)























[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice Properties

Public Properties

	Name	Description
	ClipPlanes	Retrieves the clipping planes of the current GraphicsDevice.
	CreationParameters	Retrieves the creation parameters of the GraphicsDevice.
	DepthStencilBuffer	Gets or sets the depth stencil surface of this GraphicsDevice .
	DisplayMode	Retrieves the display mode's spatial resolution, color resolution, and refresh frequency.
	DriverLevel	Returns the driver level.
	GraphicsDeviceCapabilities	Gets the capabilities of the graphics device.
	GraphicsDeviceStatus	Retrieves the status of the device
	Indices	Gets or sets index data.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	PixelShader	Gets or sets the current pixel shader.
	PresentationParameters	Gets the presentation parameters associated with this graphics device.
	RasterStatus	Retrieves information that describes the raster of the monitor on which the swap chain is presented.
	RenderState	Retrieves a render-state value for a GraphicsDevice.
	SamplerStates	Retrieves a collection of SamplerState objects for the current GraphicsDevice .
	ScissorRectangle	Gets or sets the rectangle used for scissor testing.
	Textures	Returns the collection of textures that have been assigned to the texture stages of the device.
	VertexDeclaration	Gets or sets a vertex shader declaration.
	VertexSamplerStates	Gets the collection of vertex sampler states.
	VertexShader	Gets or sets the current vertex shader.
	VertexTextures	Gets the collection of vertex textures that support texture lookup in the vertex shader using the texldl - vs texture load statement. The vertex engine contains four texture sampler stages.
	Vertices	Gets the vertex stream collection.
	Viewport	Gets or sets a viewport identifying the portion of the render target to receive draw calls.

See Also

Reference

[GraphicsDevice Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.ClipPlanes Property

Retrieves the clipping planes of the current GraphicsDevice.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ClipPlaneCollection ClipPlanes { get; }
```

Property Value

The clipping planes.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.CreationParameters Property

Retrieves the creation parameters of the GraphicsDevice.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDeviceCreationParameters CreationParameters { get; }
```

Property Value

The creation parameters.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.DepthStencilBuffer Property

Gets or sets the depth stencil surface of this [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DepthStencilBuffer DepthStencilBuffer { get; set; }
```

Property Value

The depth stencil buffer. If no depth stencil buffer is associated with the [GraphicsDevice](#), the return value is **null**.

Setting this value to **null** disables the depth stencil operation.

Exceptions

Exception type	Condition
ObjectDisposedException	DepthStencilBuffer was called after the GraphicsDevice was disposed.

Remarks

The following restrictions apply when using this property.

- The multisample type must be the same for the render target and the depth stencil surface.
- The formats for the render target and depth stencil surface must be compatible. For more information, see [CheckDepthStencilMatch](#).
- The size of the depth stencil surface must be greater than or equal to the size of the render target.

These restrictions are validated only when using the debug runtime when any of the [GraphicsDevice](#) drawing methods are called.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.DisplayMode Property

Retrieves the display mode's spatial resolution, color resolution, and refresh frequency.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DisplayMode DisplayMode { get; }
```

Property Value

Data describing the display mode of the adapter (as opposed to the display mode of the [GraphicsDevice](#), which might not be active if the [GraphicsDevice](#) does not own full-screen mode).

Remarks For a list of all supported display modes for an adapter, see the [SupportedDisplayModes](#) property.

Example

This simple example checks for a supported display mode and sets the back buffer dimensions to match the resolution of the desired display mode.

C#

```
public class Game1 : Microsoft.Xna.Framework.Game
{
    GraphicsDeviceManager graphics;

    public Game1()
    {
        graphics = new GraphicsDeviceManager(this);
        Content.RootDirectory = "Content";

        graphics.PreparingDeviceSettings +=
            new EventHandler<PreparingDeviceSettingsEventArgs>(
                graphics_PreparingDeviceSettings);
    }

    /// <summary>
    /// Modifies the display mode for the graphics device
    /// when it is reset or recreated.
    /// </summary>
    void graphics_PreparingDeviceSettings(object sender,
        PreparingDeviceSettingsEventArgs e)
    {
        foreach (Microsoft.Xna.Framework.Graphics.DisplayMode displayMode
            in GraphicsAdapter.DefaultAdapter.SupportedDisplayModes)
        {
            // If 1080i (Widescreen) format is available,
            // set it to use this format.
            // 1080i is 1920x1080, so check to see if there is a
            // DisplayMode available that matches.
            if (displayMode.Width == 1920 && displayMode.Height == 1080)
            {
                e.GraphicsDeviceInformation.PresentationParameters.
                    BackBufferFormat = displayMode.Format;
                e.GraphicsDeviceInformation.PresentationParameters.
                    BackBufferHeight = displayMode.Height;
                e.GraphicsDeviceInformation.PresentationParameters.
                    BackBufferWidth = displayMode.Width;
                e.GraphicsDeviceInformation.PresentationParameters.
                    FullScreenRefreshRateInHz = displayMode.RefreshRate;
                e.GraphicsDeviceInformation.PresentationParameters.
                    IsFullScreen = true;
            }
        }
    }
}
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

[Xbox 360 Programming Considerations](#)

Reference

[GraphicsAdapter.SupportedDisplayModes Property](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

[Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune](#)

GraphicsDevice.DriverLevel Property

Returns the driver level.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int DriverLevel { get; }
```

Property Value

This method returns the driver version, which is one of the following:

- 700 - Direct3D 7 level driver
- 800 - Direct3D 8 level driver
- 900 - Direct3D 9 level driver

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.GraphicsDeviceCapabilities Property

Gets the capabilities of the graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDeviceCapabilities GraphicsDeviceCapabilities { get; }
```

Property Value

The capabilities.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.GraphicsDeviceStatus Property

Retrieves the status of the device

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDeviceStatus GraphicsDeviceStatus { get; }
```

Property Value

The status of the device

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Indices Property

Gets or sets index data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IndexBuffer Indices { get; set; }
```

Property Value

The index data.

Exceptions

Exception type	Condition
ObjectDisposedException	Indices was called after this GraphicsDevice was disposed.

Example

The vertex stream and index data of the graphics device must be set before any call to [DrawIndexedPrimitives](#). The following example sets the index data and associates a user-created vertex buffer of type **VertexPositionNormalTexture** with vertex stream 0 (zero) of the graphics device.

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);  
  
graphics.GraphicsDevice.Indices = lineListIndexBuffer;
```

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.PixelShader Property

Gets or sets the current pixel shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PixelShader PixelShader { get; set; }
```

Property Value

The current pixel shader or a pixel shader object to set.

Exceptions

Exception type	Condition
ObjectDisposedException	PixelShader was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.PresentationParameters Property

Gets the presentation parameters associated with this graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PresentationParameters PresentationParameters { get; }
```

Property Value

The presentation parameters associated with this graphics device.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.RasterStatus Property

Retrieves information that describes the raster of the monitor on which the swap chain is presented.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RasterStatus RasterStatus { get; }
```

Property Value

Information about the position or other status of the raster on the monitor driven by the current adapter.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.RenderState Property

Retrieves a render-state value for a GraphicsDevice.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderState RenderState { get; }
```

Property Value

The render state.

Remarks This method returns the last render state that was set for the GraphicsDevice, or the default GraphicsDevice render state, if not previously set.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.SamplerStates Property

Retrieves a collection of [SamplerState](#) objects for the current [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SamplerStateCollection SamplerStates { get; }
```

Property Value

The sample states of this [GraphicsDevice](#).

Remarks

This method returns the last sampler state that was set for the [GraphicsDevice](#), or the default [GraphicsDevice](#) sampler state, if not previously set. Programmable shaders reference textures using the sampler number, which is set as the index of a [GraphicsDevice.Textures](#).

See Also

Reference

[GraphicsDevice.Textures Property](#)

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.ScissorRectangle Property

Gets or sets the rectangle used for scissor testing.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rectangle ScissorRectangle { get; set; }
```

Property Value

Defines the rendering area within the render target, if scissor testing is enabled.

RemarksThe scissor rectangle is used as a rectangular clipping region. Use [ScissorTestEnable](#) to enable scissor testing.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Textures Property

Returns the collection of textures that have been assigned to the texture stages of the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCollection Textures { get; }
```

Property Value

The texture collection.

RemarksThe collection of texture stage on the device contains one or more textures (all of the same type and dimensions) that can be accessed by shaders.

Note

At draw time, a texture cannot be simultaneously set as a render target and a texture at a stage.

Example

This C# code demonstrates the setting two elements of the **Textures** collection to two user-created [Texture2D](#) objects named **firstTexture** and **secondTexture**.

C#

```
graphics.GraphicsDevice.Textures[0] = firstTexture;  
graphics.GraphicsDevice.Textures[1] = secondTexture;
```

In the effect file, these textures can be accessed by declaring two sampler variables and setting them to the sampler register numbers that correspond to the index in the texture collection.

```
sampler firstSampler : register(s0);  
sampler secondSampler : register(s1);
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.VertexDeclaration Property

Gets or sets a vertex shader declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexDeclaration VertexDeclaration { get; set; }
```

Property Value

The vertex shader declaration.

Exceptions

Exception type	Condition
ObjectDisposedException	VertexDeclaration was called after this GraphicsDevice was disposed.

RemarksA vertex shader declaration is made up of an array of vertex elements.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.VertexSamplerStates Property

Gets the collection of vertex sampler states.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SamplerStateCollection VertexSamplerStates { get; }
```

Property Value

The collection of vertex sampler states.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.VertexShader Property

Gets or sets the current vertex shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexShader VertexShader { get; set; }
```

Property Value

The [GraphicsDevice](#)'s current vertex shader or a vertex shader object to set.

Exceptions

Exception type	Condition
ObjectDisposedException	VertexShader was called after this GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.VertexTextures Property

Gets the collection of vertex textures that support texture lookup in the vertex shader using the **texldl** - vs texture load statement. The vertex engine contains four texture sampler stages.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCollection VertexTextures { get; }
```

Property Value

The collection of vertex textures.

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.Vertices Property

Gets the vertex stream collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexStreamCollection Vertices { get; }
```

Property Value

The vertex stream collection.

Example

The vertex stream of the graphics device must be set before any call to [DrawPrimitives](#). The following example associates a user created vertex buffer of type **VertexPositionNormalTexture** with vertex stream 0 (zero) of the graphics device.

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);
```

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

GraphicsDevice.Viewport Property

Gets or sets a viewport identifying the portion of the render target to receive draw calls.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Viewport Viewport { get; set; }
```

Property Value

The viewport to set or get.

Remarks

Viewport can be used to draw on part of the screen. It should be set before any geometry is drawn so the viewport parameters will take effect.

To draw multiple views within a scene, repeat setting **Viewport** and draw a geometry sequence for each view.

Example

This code sample, from [How To: Use Viewports for Split Screen Gaming](#), demonstrates how to use the **Viewport** property to display different scenes to different parts of the screen.

C#

```
Viewport defaultViewport;
Viewport leftViewport;
Viewport rightViewport;
Matrix projectionMatrix;
Matrix halfprojectionMatrix;
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    defaultViewport = GraphicsDevice.Viewport;
    leftViewport = defaultViewport;
    rightViewport = defaultViewport;
    leftViewport.Width = leftViewport.Width / 2;
    rightViewport.Width = rightViewport.Width / 2;
    rightViewport.X = leftViewport.Width + 1;

    Ring = Content.Load<Model>("redtorus");

    projectionMatrix = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.PiOver4, 4.0f / 3.0f, 1.0f, 10000f);
    halfprojectionMatrix = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.PiOver4, 2.0f / 3.0f, 1.0f, 10000f);
}
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Viewport = defaultViewport;
    GraphicsDevice.Clear(Color.CornflowerBlue);

    GraphicsDevice.Viewport = leftViewport;
    DrawScene(gameTime, Camera1.ViewMatrix, halfprojectionMatrix);
    GraphicsDevice.Viewport = rightViewport;
    DrawScene(gameTime, Camera2.ViewMatrix, halfprojectionMatrix);

    base.Draw(gameTime);
}
```

See Also

Tasks

[How To: Use Viewports for Split Screen Gaming](#)

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[GraphicsDevice Class](#)







[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice Events

Public Events

	Name	Description
	DeviceLost	Occurs when a GraphicsDevice is about to be lost (for example, immediately before a reset).
	DeviceReset	Occurs after a GraphicsDevice is reset, allowing an application to recreate all resources.
	DeviceResetting	Occurs when a GraphicsDevice is resetting, allowing the application to cancel the default handling of the reset.
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.
	ResourceCreated	Occurs when a resource is created.
	ResourceDestroyed	Occurs when a resource is destroyed.

See Also

Reference

[GraphicsDevice Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDevice.DeviceLost Event

Occurs when a GraphicsDevice is about to be lost (for example, immediately before a reset).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler DeviceLost
```

Example

To add an event handler that listens for **DeviceLost**, use the following C# code.

```
[C#]
GraphicsDevice.DeviceLost += new System.EventHandler( this.GraphicsDeviceLostEven
tHandler );
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.DeviceReset Event

Occurs after a GraphicsDevice is reset, allowing an application to recreate all resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler DeviceReset
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.DeviceResetting Event

Occurs when a GraphicsDevice is resetting, allowing the application to cancel the default handling of the reset.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler DeviceResetting
```

Example

To add an event handler that listens for **DeviceResetting**, use the following C# code.

```
[C#]
GraphicsDevice.DeviceResetting += new System.CancelEventHandler( this.GraphicsDev
iceResettingEventHandler );
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

Remarks [IsDisposed](#) indicates whether an object has been disposed.

Example

To add an event handler that listens for **Disposing**, use the following C# code.

```
[C#]  
obj.Disposing += new System.EventHandler( this.OnDisposing );
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.ResourceCreated Event

Occurs when a resource is created.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<ResourceCreatedEventArgs> ResourceCreated
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDevice.ResourceDestroyed Event

Occurs when a resource is destroyed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<ResourceDestroyedEventArgs> ResourceDestroyed
```

See Also

Reference

[GraphicsDevice Class](#)

[GraphicsDevice Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities Class

Represents the capabilities of the hardware.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class GraphicsDeviceCapabilities : IDisposable
```

See Also

Tasks

[How To: Check for Shader Model 2.0 Support](#)

Reference

[GraphicsDeviceCapabilities Members](#)



















[Microsoft.Xna.Framework.Graphics Namespace](#)



















Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities Members









The following tables list the members exposed by the GraphicsDeviceCapabilities type.

Public Properties



Name	Description
 AdapterOrdinalInGroup	Gets the number indicating the order in which heads are referenced by the API.
 AlphaCompareCapabilities	Gets the alpha-test comparison capabilities.
 CubeTextureFilterCapabilities	Gets texture-filtering capabilities for a cube texture.
 CursorCapabilities	Gets hardware cursor capabilities.
 DeclarationTypeCapabilities	Gets vertex data type capabilities.
 DepthBufferCompareCapabilities	Gets depth buffer comparison capabilities.
 DestinationBlendCapabilities	Gets destination-blending capabilities.
 DeviceCapabilities	Gets graphics device capabilities.
 DeviceType	Returns the type of the current GraphicsDevice .
 DriverCapabilities	Gets device driver capabilities.
 ExtentsAdjust	Gets the number of pixels to adjust the extents rectangle outward to accommodate antialiasing kernels.
 GuardBandBottom	Gets the bottom screen-space coordinate of the guard-band clipping region.
 GuardBandLeft	Gets the left screen-space coordinate of the guard-band clipping region.
 GuardBandRight	Gets the right coordinate of the guard-band clipping region.
 GuardBandTop	Gets the top screen-space coordinate of the guard-band clipping region.
 LineCapabilities	Gets line-drawing primitives capabilities.
 MasterAdapterOrdinal	Gets the master device for this subordinate.
 MaxAnisotropy	Gets the maximum valid value for the MaxAnisotropy texture-stage state.
 MaxPixelShader30InstructionSlots	Gets the maximum number of pixel shader instruction slots supported.
 MaxPixelShaderProfile	Returns the maximum pixel shader profile for this GraphicsDevice .
 MaxPointSize	Gets the maximum size of a point primitive.
 MaxPrimitiveCount	Gets the maximum number of primitives for each DrawPrimitives call.
 MaxSimultaneousRenderTarget	Gets the maximum number of simultaneous render targets.
 MaxSimultaneousTextures	Gets the maximum number of textures that can be simultaneously bound to the fixed-function pipeline sampler stages.
 MaxStreams	Gets the maximum number of concurrent data streams.
 MaxStreamStride	Gets the maximum stride.
 MaxTextureAspectRatio	Gets the maximum texture aspect ratio supported by the hardware.
 MaxTextureHeight	Gets the maximum texture height for this device.
 MaxTextureRepeat	Gets the maximum range of the integer bits of the post-normalized texture coordinates.
 MaxTextureWidth	Gets the maximum texture width for this device.
 MaxUserClipPlanes	Gets the maximum number of user-defined clipping planes supported.
 MaxVertexIndex	Gets the maximum size of indices supported for hardware vertex processing.
 MaxVertexShader30InstructionSlots	Gets the maximum number of vertex shader instruction slots supported.
 MaxVertexShaderConstants	Gets the number of vertex shader registers that are reserved for constants.
 MaxVertexShaderProfile	Returns the maximum vertex shader profile for this GraphicsDevice .
 MaxVertexW	Gets the maximum W-based depth value that the device supports.
 MaxVolumeExtent	Gets the maximum value for any of the three dimensions (width, height, and depth) of a volume texture.
 NumberOfAdaptersInGroup	Gets the number of adapters in this adapter group (only if master).
 PixelShader1xMaxValue	Gets the maximum value of pixel shader arithmetic component.

 PixelShaderCapabilities	Gets the pixel shader 2.0 capabilities.
 PixelShaderVersion	Gets the pixel shader main and sub versions.
 PresentInterval	Gets the rate at which the swap chain's back buffers are presented to the front buffer.
 PrimitiveCapabilities	Gets driver primitive capabilities.
 RasterCapabilities	Gets information on raster-drawing capabilities.
 ShadingCapabilities	Gets shading operations capabilities.
 SourceBlendCapabilities	Gets source-blending capabilities.
 StencilCapabilities	Gets the supported stencil-buffer operations.
 TextureAddressCapabilities	Gets texture-addressing capabilities for texture objects.
 TextureCapabilities	Gets miscellaneous texture-mapping capabilities.
 TextureFilterCapabilities	Gets texture-filtering capabilities for a texture.
 VertexFormatCapabilities	Gets flexible vertex format capabilities.
 VertexProcessingCapabilities	Gets vertex processing capabilities.
 VertexShaderCapabilities	Gets vertex shader version 2.0 extended capabilities.
 VertexShaderVersion	Gets the vertex shader main and sub versions.
 VertexTextureFilterCapabilities	Gets vertex shader texture filter capabilities.
 VolumeTextureAddressCapabilities	Gets texture-addressing capabilities for a volume texture
 VolumeTextureFilterCapabilities	Gets texture-filtering capabilities for a volume texture.

Public Methods

	Name	Description
	Dispose	Overloaded. Releases all resources used by the GraphicsDeviceCapabilities class.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares two objects to determine whether they are the same.
	Op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[GraphicsDeviceCapabilities Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Releases all resources used by the GraphicsDeviceCapabilities class.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares two objects to determine whether they are the same.
	 op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.Dispose Method

Releases all resources used by the [GraphicsDeviceCapabilities](#) class.

Overload List

Name	Description
GraphicsDeviceCapabilities.Dispose ()	Releases all resources used by the GraphicsDeviceCapabilities class.
GraphicsDeviceCapabilities.Dispose (Boolean)	Releases the unmanaged resources used by the GraphicsDeviceCapabilities object and optionally releases the managed resources.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.Dispose Method ()

Releases all resources used by the [GraphicsDeviceCapabilities](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [GraphicsDeviceCapabilities](#). The [Dispose](#) method leaves the [GraphicsDeviceCapabilities](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GraphicsDeviceCapabilities](#) so the garbage collector can reclaim the memory that the [GraphicsDeviceCapabilities](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GraphicsDeviceCapabilities](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GraphicsDeviceCapabilities](#) object's [Finalize](#) method.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.Dispose Method (Boolean)

Note

This method is available only when developing for Windows.

Releases the unmanaged resources used by the [GraphicsDeviceCapabilities](#) object and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDeviceCapabilities.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GraphicsDeviceCapabilities.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GraphicsDeviceCapabilities.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [GraphicsDeviceCapabilities](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GraphicsDeviceCapabilities left,  
    GraphicsDeviceCapabilities right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GraphicsDeviceCapabilities left,  
    GraphicsDeviceCapabilities right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)










































[GraphicsDeviceCapabilities Members](#)

















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities Properties

Public Properties

Name	Description
 AdapterOrdinalInGroup	Gets the number indicating the order in which heads are referenced by the API.
 AlphaCompareCapabilities	Gets the alpha-test comparison capabilities.
 CubeTextureFilterCapabilities	Gets texture-filtering capabilities for a cube texture.
 CursorCapabilities	Gets hardware cursor capabilities.
 DeclarationTypeCapabilities	Gets vertex data type capabilities.
 DepthBufferCompareCapabilities	Gets depth buffer comparison capabilities.
 DestinationBlendCapabilities	Gets destination-blending capabilities.
 DeviceCapabilities	Gets graphics device capabilities.
 DeviceType	Returns the type of the current GraphicsDevice .
 DriverCapabilities	Gets device driver capabilities.
 ExtentsAdjust	Gets the number of pixels to adjust the extents rectangle outward to accommodate antialiasing kernels.
 GuardBandBottom	Gets the bottom screen-space coordinate of the guard-band clipping region.
 GuardBandLeft	Gets the left screen-space coordinate of the guard-band clipping region.
 GuardBandRight	Gets the right coordinate of the guard-band clipping region.
 GuardBandTop	Gets the top screen-space coordinate of the guard-band clipping region.
 LineCapabilities	Gets line-drawing primitives capabilities.
 MasterAdapterOrdinal	Gets the master device for this subordinate.
 MaxAnisotropy	Gets the maximum valid value for the MaxAnisotropy texture-stage state.
 MaxPixelShader30InstructionSlots	Gets the maximum number of pixel shader instruction slots supported.
 MaxPixelShaderProfile	Returns the maximum pixel shader profile for this GraphicsDevice .
 MaxPointSize	Gets the maximum size of a point primitive.
 MaxPrimitiveCount	Gets the maximum number of primitives for each DrawPrimitives call.
 MaxSimultaneousRenderTarget	Gets the maximum number of simultaneous render targets.
 MaxSimultaneousTextures	Gets the maximum number of textures that can be simultaneously bound to the fixed-function pipeline sampler stages.
 MaxStreams	Gets the maximum number of concurrent data streams.
 MaxStreamStride	Gets the maximum stride.
 MaxTextureAspectRatio	Gets the maximum texture aspect ratio supported by the hardware.
 MaxTextureHeight	Gets the maximum texture height for this device.
 MaxTextureRepeat	Gets the maximum range of the integer bits of the post-normalized texture coordinates.
 MaxTextureWidth	Gets the maximum texture width for this device.
 MaxUserClipPlanes	Gets the maximum number of user-defined clipping planes supported.
 MaxVertexIndex	Gets the maximum size of indices supported for hardware vertex processing.
 MaxVertexShader30InstructionSlots	Gets the maximum number of vertex shader instruction slots supported.
 MaxVertexShaderConstants	Gets the number of vertex shader registers that are reserved for constants.
 MaxVertexShaderProfile	Returns the maximum vertex shader profile for this GraphicsDevice .
 MaxVertexW	Gets the maximum W-based depth value that the device supports.
 MaxVolumeExtent	Gets the maximum value for any of the three dimensions (width, height, and depth) of a volume texture.
 NumberOfAdaptersInGroup	Gets the number of adapters in this adapter group (only if master).
 PixelShader1xMaxValue	Gets the maximum value of pixel shader arithmetic component.
 PixelShaderCapabilities	Gets the pixel shader 2.0 capabilities.
 PixelShaderVersion	Gets the pixel shader main and sub versions.

 PresentInterval	Gets the rate at which the swap chain's back buffers are presented to the front buffer.
 PrimitiveCapabilities	Gets driver primitive capabilities.
 RasterCapabilities	Gets information on raster-drawing capabilities.
 ShadingCapabilities	Gets shading operations capabilities.
 SourceBlendCapabilities	Gets source-blending capabilities.
 StencilCapabilities	Gets the supported stencil-buffer operations.
 TextureAddressCapabilities	Gets texture-addressing capabilities for texture objects.
 TextureCapabilities	Gets miscellaneous texture-mapping capabilities.
 TextureFilterCapabilities	Gets texture-filtering capabilities for a texture.
 VertexFormatCapabilities	Gets flexible vertex format capabilities.
 VertexProcessingCapabilities	Gets vertex processing capabilities.
 VertexShaderCapabilities	Gets vertex shader version 2.0 extended capabilities.
 VertexShaderVersion	Gets the vertex shader main and sub versions.
 VertexTextureFilterCapabilities	Gets vertex shader texture filter capabilities.
 VolumeTextureAddressCapabilities	Gets texture-addressing capabilities for a volume texture
 VolumeTextureFilterCapabilities	Gets texture-filtering capabilities for a volume texture.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.AdapterOrdinalInGroup Property

Gets the number indicating the order in which heads are referenced by the API.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int AdapterOrdinalInGroup { get; }
```

Property Value

The number indicating the order in which heads are referenced by the API.

Remarks

The value for the master adapter is always 0. These values do not correspond to the adapter ordinals. They apply only to heads within a group.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AlphaCompareCapabilities Property

Gets the alpha-test comparison capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompareCaps AlphaCompareCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.CompareCaps](#) representing the alpha-test comparison capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CubeTextureFilterCapabilities Property

Gets texture-filtering capabilities for a cube texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FilterCaps CubeTextureFilterCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.FilterCaps](#) representing the texture-filtering capabilities for a cube texture of a device.

Remarks

Per-stage filtering capabilities reflect which filtering modes are supported for texture stages when performing multiple-texture blending. This member can be any combination of the per-stage texture-filtering flags defined in

[GraphicsDeviceCapabilities.FilterCaps](#).

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CursorCapabilities Property

Gets hardware cursor capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CursorCaps CursorCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.CursorCaps](#) representing the hardware cursor capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCapabilities Property

Gets vertex data type capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeclarationTypeCaps DeclarationTypeCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.DeclarationTypeCaps](#) representing the vertex data type capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DepthBufferCompareCapabilities Property

Gets depth buffer comparison capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompareCaps DepthBufferCompareCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.CompareCaps](#) representing the z-buffer comparison capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DestinationBlendCapabilities Property

Gets destination-blending capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BlendCaps DestinationBlendCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.BlendCaps](#) representing the destination-blending capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCapabilities Property

Gets graphics device capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceCaps DeviceCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.DeviceCaps](#) representing the graphics capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceType Property

Returns the type of the current [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceType DeviceType { get; }
```

Property Value

The type of the current [GraphicsDevice](#).

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCapabilities Property

Gets device driver capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DriverCaps DriverCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.DriverCaps](#) representing the driver capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ExtentsAdjust Property

Gets the number of pixels to adjust the extents rectangle outward to accommodate antialiasing kernels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float ExtentsAdjust { get; }
```

Property Value

Size of adjustment, in pixels.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.GuardBandBottom Property

Gets the bottom screen-space coordinate of the guard-band clipping region.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GuardBandBottom { get; }
```

Property Value

The bottom coordinate of the guard-band clipping region.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.GuardBandLeft Property

Gets the left screen-space coordinate of the guard-band clipping region.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GuardBandLeft { get; }
```

Property Value

The left coordinate of the guard-band clipping region.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.GuardBandRight Property

Gets the right coordinate of the guard-band clipping region.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GuardBandRight { get; }
```

Property Value

The right coordinate of the guard-band clipping region.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.GuardBandTop Property

Gets the top screen-space coordinate of the guard-band clipping region.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float GuardBandTop { get; }
```

Property Value

The top coordinate of the guard-band clipping region.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCapabilities Property

Gets line-drawing primitives capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public LineCaps LineCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.LineCaps](#) representing the line-drawing primitives capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MasterAdapterOrdinal Property

Gets the master device for this subordinate.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MasterAdapterOrdinal { get; }
```

Property Value

Number indicating the master device. This number is taken from the same space as the adapter values.

Remarks

For multi-head support, one head will be denoted the master head, and all other heads on the same card will be denoted subordinate heads. If more than one multi-head adapter is present in a system, the master and its subordinates from one multi-head adapter are called a group.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxAnisotropy Property

Gets the maximum valid value for the [MaxAnisotropy](#) texture-stage state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxAnisotropy { get; }
```

Property Value

The maximum valid value for [MaxAnisotropy](#).

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxPixelShader30InstructionSlots Property

Gets the maximum number of pixel shader instruction slots supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxPixelShader30InstructionSlots { get; }
```

Property Value

The maximum number of pixel shader instruction slots supported.

Remarks

The maximum value that can be set on this capability is 32,768. Devices that support ps_3_0 are required to support at least 512 instruction slots.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxPixelShaderProfile Property

Returns the maximum pixel shader profile for this [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderProfile MaxPixelShaderProfile { get; }
```

Property Value

The maximum pixel shader profile for this [GraphicsDevice](#).

See Also

Tasks

[How To: Check for Shader Model 2.0 Support](#)

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxPointSize Property

Gets the maximum size of a point primitive.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MaxPointSize { get; }
```

Property Value

The maximum size of a point primitive. The range is greater than or equal to 1.0f.

Remarks

If the size is set to 1.0f, the device does not support point size control.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxPrimitiveCount Property

Gets the maximum number of primitives for each [DrawPrimitives](#) call.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxPrimitiveCount { get; }
```

Property Value

The maximum number of primitives.

See Also

Reference

[DrawPrimitives](#)

[DrawIndexedPrimitives](#)

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxSimultaneousRenderTargets Property

Gets the maximum number of simultaneous render targets.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxSimultaneousRenderTargets { get; }
```

Property Value

The maximum number of simultaneous render targets supported by the device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxSimultaneousTextures Property

Gets the maximum number of textures that can be simultaneously bound to the fixed-function pipeline sampler stages.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxSimultaneousTextures { get; }
```

Property Value

The maximum number of textures that can be simultaneously bound to the fixed-function pipeline sampler stages.

Remarks

If the same texture is bound to two sampler stages, it counts as two textures.

This value has no meaning in the programmable pipeline where the number of sampler stages is determined by each pixel shader version. Each pixel shader version also determines the number of texture declaration instructions.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxStreams Property

Gets the maximum number of concurrent data streams.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxStreams { get; }
```

Property Value

The maximum number of concurrent data streams. The valid range is 1 to 16.

Remarks

If this value is 0, then the driver is not a Direct3D 9 driver.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxStreamStride Property

Gets the maximum stride.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxStreamStride { get; }
```

Property Value

The maximum stride.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxTextureAspectRatio Property

Gets the maximum texture aspect ratio supported by the hardware.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxTextureAspectRatio { get; }
```

Property Value

The maximum texture aspect ratio supported by the hardware, typically a power of 2.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxTextureHeight Property

Gets the maximum texture height for this device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxTextureHeight { get; }
```

Property Value

The maximum texture height for this device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxTextureRepeat Property

Gets the maximum range of the integer bits of the post-normalized texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxTextureRepeat { get; }
```

Property Value

The maximum range of the integer bits of the post-normalized texture coordinates.

Remarks

A texture coordinate is stored as a 32-bit signed integer using 27 bits to store the integer part and 5 bits for the floating-point fraction. The maximum integer index, 227, is used to determine the maximum texture coordinate, depending on how the hardware does texture-coordinate scaling.

Some hardware reports the cap [TextureCaps.SupportsTextureRepeatNotScaledBySize](#). For this case, the device defers scaling texture coordinates by the texture size until after interpolation and application of the texture address mode, so the number of times a texture can be wrapped is given by the integer value in **MaxTextureRepeat**.

Less desirably, on some hardware [TextureCaps.SupportsTextureRepeatNotScaledBySize](#) is not set and the device scales the texture coordinates by the texture size (using the highest level of detail) prior to interpolation. This limits the number of times a texture can be wrapped to **MaxTextureRepeat** / texture size.

For example, assume that **MaxTextureRepeat** is equal to 32K and the size of the texture is 4K. If the hardware sets [TextureCaps.SupportsTextureRepeatNotScaledBySize](#), then the number of times a texture can be wrapped is equal to **MaxTextureRepeat** (32K, in this example). Otherwise, the number of times a texture can be wrapped is equal to **MaxTextureRepeat** divided by texture size, which, in this example, is 32K/4K, or 8.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxTextureWidth Property

Gets the maximum texture width for this device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxTextureWidth { get; }
```

Property Value

The maximum texture width for this device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxUserClipPlanes Property

Gets the maximum number of user-defined clipping planes supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxUserClipPlanes { get; }
```

Property Value

The maximum number of user-defined clipping planes supported. This number can be 0.

Remarks

For a given physical device, this capability might vary across Direct3D devices, depending on the parameters supplied to the [Device](#) constructor.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxVertexIndex Property

Gets the maximum size of indices supported for hardware vertex processing.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxVertexIndex { get; }
```

Property Value

The maximum size of indices supported for hardware vertex processing.

Remarks

It is possible to create 32-bit index buffers; however, you will not be able to render with the index buffer unless this value is greater than 0x0000FFFF.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxVertexShader30InstructionSlots Property

Gets the maximum number of vertex shader instruction slots supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxVertexShader30InstructionSlots { get; }
```

Property Value

The maximum number of vertex shader instruction slots supported.

Remarks

The maximum value that can be set on this cap is 32,768. Devices that support vs_3_0 are required to support at least 512 instruction slots.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxVertexShaderConstants

Property

Gets the number of vertex shader registers that are reserved for constants.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxVertexShaderConstants { get; }
```

Property Value

The number of vertex shader registers that are reserved for constants.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxVertexShaderProfile Property

Returns the maximum vertex shader profile for this [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderProfile MaxVertexShaderProfile { get; }
```

Property Value

The maximum vertex shader profile for this [GraphicsDevice](#).

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxVertexW Property

Gets the maximum W-based depth value that the device supports.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MaxVertexW { get; }
```

Property Value

The maximum W-based depth value that the device supports.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.MaxVolumeExtent Property

Gets the maximum value for any of the three dimensions (width, height, and depth) of a volume texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxVolumeExtent { get; }
```

Property Value

The maximum value for any of the three dimensions (width, height, and depth) of a volume texture.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.NumberOfAdaptersInGroup Property

Gets the number of adapters in this adapter group (only if master).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int NumberOfAdaptersInGroup { get; }
```

Property Value

The number of adapters in this adapter group (only if master).

Remarks

NumberOfAdaptersInGroup will be 1 for conventional adapters, greater than 1 for the master adapter of a multi-head card, or 0 for a subordinate adapter of a multihead card. Each card can have at most one master, but may have many subordinates.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShader1xMaxValue Property

Gets the maximum value of pixel shader arithmetic component.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float PixelShader1xMaxValue { get; }
```

Property Value

The maximum value of pixel shader arithmetic component.

Remarks

PixelShader1xMaxValue indicates the internal range of values supported for pixel color blending operations. Within the range that they report to, implementations must allow data to pass through pixel processing unmodified (unclamped). Normally, the value of this member is an absolute value. For example, a 1.0 indicates that the range is -1.0 to 1 , and an 8.0 indicates that the range is -8.0 to 8.0 . The value must be greater than or equal to 1.0 for any hardware that supports pixel shaders.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCapabilities Property

Gets the pixel shader 2.0 capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PixelShaderCaps PixelShaderCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.PixelShaderCaps](#) representing the pixel shader 2.0 capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderVersion Property

Gets the pixel shader main and sub versions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Version PixelShaderVersion { get; }
```

Property Value

The pixel shader main and sub versions.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PresentInterval Property

Gets the rate at which the swap chain's back buffers are presented to the front buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PresentInterval PresentInterval { get; }
```

Property Value

The rate at which the swap chain's back buffers are presented to the front buffer.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCapabilities Property

Gets driver primitive capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PrimitiveCaps PrimitiveCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.PrimitiveCaps](#) representing the primitive capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCapabilities Property

Gets information on raster-drawing capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RasterCaps RasterCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.RasterCaps](#) representing the raster capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCapabilities Property

Gets shading operations capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShadingCaps ShadingCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.ShadingCaps](#) representing the shading capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.SourceBlendCapabilities Property

Gets source-blending capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BlendCaps SourceBlendCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.BlendCaps](#) representing the source-blending capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCapabilities Property

Gets the supported stencil-buffer operations.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilCaps StencilCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.StencilCaps](#) representing the stencil-buffer capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureAddressCapabilities Property

Gets texture-addressing capabilities for texture objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AddressCaps TextureAddressCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.AddressCaps](#) representing the texture-addressing capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCapabilities Property

Gets miscellaneous texture-mapping capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCaps TextureCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.TextureCaps](#) representing the texture-mapping capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureFilterCapabilities Property

Gets texture-filtering capabilities for a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FilterCaps TextureFilterCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.FilterCaps](#) representing the texture-filtering capabilities of a device.

Remarks

Per-stage filtering capabilities reflect which filtering modes are supported for texture stages when performing multiple-texture blending. This member can be any combination of the per-stage texture-filtering flags defined in [GraphicsDeviceCapabilities.FilterCaps](#).

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCapabilities Property

Gets flexible vertex format capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexFormatCaps VertexFormatCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.VertexFormatCaps](#) representing the flexible vertex format capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCapabilities Property

Gets vertex processing capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexProcessingCaps VertexProcessingCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.VertexProcessingCaps](#) representing the vertex processing capabilities of a device.

Remarks

For a given physical device, this capability might vary across Direct3D devices, depending on the parameters supplied to the [Device](#) constructor.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCapabilities Property

Gets vertex shader version 2.0 extended capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexShaderCaps VertexShaderCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.VertexShaderCaps](#) representing the vertex shader 2.0 capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderVersion Property

Gets the vertex shader main and sub versions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Version VertexShaderVersion { get; }
```

Property Value

The vertex shader main and sub versions.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexTextureFilterCapabilities Property

Gets vertex shader texture filter capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FilterCaps VertexTextureFilterCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.FilterCaps](#) representing the vertex shader filter capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VolumeTextureAddressCapabilities Property

Gets texture-addressing capabilities for a volume texture

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AddressCaps VolumeTextureAddressCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.AddressCaps](#) representing the texture-addressing capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VolumeTextureFilterCapabilities Property

Gets texture-filtering capabilities for a volume texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FilterCaps VolumeTextureFilterCapabilities { get; }
```

Property Value

A [GraphicsDeviceCapabilities.FilterCaps](#) representing the texture-filtering capabilities of a device.

See Also

Reference

[GraphicsDeviceCapabilities Class](#)

[GraphicsDeviceCapabilities Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps Structure

Represents the texture addressing capabilities for [Texture](#) structures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.AddressCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)







[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.AddressCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.AddressCaps type.



Public Properties

Name	Description
 SupportsBorder	Gets a value indicating whether the device supports the setting of coordinates outside the range [0.0, 1.0] to the border color.
 SupportsClamp	Gets a value indicating whether the device supports the clamping of textures to addresses.
 SupportsIndependentUV	Gets a value indicating whether the device can separate the texture-addressing modes of the texture's u and v coordinates.
 SupportsMirror	Gets a value indicating whether a device can mirror textures to addresses.
 SupportsMirrorOnce	Gets a value indicating whether a device can take the absolute value of the texture coordinate (thus, mirroring around 0) and then clamp to the maximum value.
 SupportsWrap	Gets a value indicating whether a device can wrap textures to addresses.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.AddressCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.AddressCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.AddressCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.AddressCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.AddressCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.AddressCaps instances are equal.
GraphicsDeviceCapabilities.AddressCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.AddressCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.AddressCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.AddressCaps](#) object this instance is being compared to .

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.AddressCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    AddressCaps left,  
    AddressCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.AddressCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.AddressCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.AddressCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.AddressCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    AddressCaps left,  
    AddressCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.AddressCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.AddressCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.AddressCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)







[GraphicsDeviceCapabilities.AddressCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps Properties

Public Properties

	Name	Description
	SupportsBorder	Gets a value indicating whether the device supports the setting of coordinates outside the range [0.0, 1.0] to the border color.
	SupportsClamp	Gets a value indicating whether the device supports the clamping of textures to addresses.
	SupportsIndependentUV	Gets a value indicating whether the device can separate the texture-addressing modes of the texture's u and v coordinates.
	SupportsMirror	Gets a value indicating whether a device can mirror textures to addresses.
	SupportsMirrorOnce	Gets a value indicating whether a device can take the absolute value of the texture coordinate (thus, mirroring around 0) and then clamp to the maximum value.
	SupportsWrap	Gets a value indicating whether a device can wrap textures to addresses.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.AddressCaps.SupportsBorder Property

Gets a value indicating whether the device supports the setting of coordinates outside the range [0.0, 1.0] to the border color.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsBorder { get; }
```

Property Value

true if the device supports the setting of coordinates outside the range to the border color; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.SupportsClamp Property

Gets a value indicating whether the device supports the clamping of textures to addresses.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsClamp { get; }
```

Property Value

true if the device supports clamping textures to addresses; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.SupportsIndependentUV Property

Gets a value indicating whether the device can separate the texture-addressing modes of the texture's u and v coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsIndependentUV { get; }
```

Property Value

true if the device supports separating the texture-addressing modes of the u and v coordinates; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.SupportsMirror Property

Gets a value indicating whether a device can mirror textures to addresses.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMirror { get; }
```

Property Value

true if the device supports mirroring textures to addresses; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.SupportsMirrorOnce Property

Gets a value indicating whether a device can take the absolute value of the texture coordinate (thus, mirroring around 0) and then clamp to the maximum value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMirrorOnce { get; }
```

Property Value

true if the device can mirror once; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.AddressCaps.SupportsWrap Property

Gets a value indicating whether a device can wrap textures to addresses.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsWrap { get; }
```

Property Value

true if the device can wrap textures to addresses; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.AddressCaps Structure](#)

[GraphicsDeviceCapabilities.AddressCaps Members](#)

[TextureAddressCapabilities](#)

[VolumeTextureAddressCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps Structure

Represents the supported blend capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.BlendCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.BlendCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.BlendCaps type.



Public Properties

Name	Description
 SupportsBlendFactor	Gets a value indicating that the driver supports the Blend.BlendFactor blend mode.
 SupportsBothInverseSourceAlpha	Gets a value indicating that the driver supports the Blend.BothInverseSourceAlpha blend mode.
 SupportsBothSourceAlpha	Gets a value indicating that the driver supports the Blend.BothSourceAlpha blend mode.
 SupportsDestinationAlpha	Gets a value indicating that the driver supports the Blend.DestinationAlpha blend mode.
 SupportsDestinationColor	Gets a value indicating that the driver supports the Blend.DestinationColor blend mode.
 SupportsInverseDestinationAlpha	Gets a value indicating that the driver supports the Blend.InverseDestinationAlpha blend mode.
 SupportsInverseDestinationColor	Gets a value indicating that the driver supports the Blend.InverseDestinationColor blend mode.
 SupportsInverseSourceAlpha	Gets a value indicating that the driver supports the Blend.InverseSourceAlpha blend mode.
 SupportsInverseSourceColor	Gets a value indicating that the driver supports the Blend.InverseSourceColor blend mode.
 SupportsOne	Gets a value indicating that the driver supports the Blend.One blend mode.
 SupportsSourceAlpha	Gets a value indicating that the driver supports the Blend.SourceAlpha blend mode.
 SupportsSourceAlphaSat	Gets a value indicating that the driver supports the Blend.SourceAlphaSat blend mode.
 SupportsSourceColor	Gets a value indicating that the driver supports the Blend.SourceColor blend mode.
 SupportsZero	Gets a value indicating that the driver supports the Blend.Zero blend mode.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.BlendCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.BlendCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.BlendCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.BlendCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.BlendCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.BlendCaps instances are equal.
GraphicsDeviceCapabilities.BlendCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.BlendCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.BlendCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.BlendCaps](#) to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.BlendCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    BlendCaps left,  
    BlendCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.BlendCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.BlendCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.BlendCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.BlendCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    BlendCaps left,  
    BlendCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.BlendCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.BlendCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.BlendCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)















[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps Properties

Public Properties

	Name	Description
	SupportsBlendFactor	Gets a value indicating that the driver supports the Blend.BlendFactor blend mode.
	SupportsBothInverseSourceAlpha	Gets a value indicating that the driver supports the Blend.BothInverseSourceAlpha blend mode.
	SupportsBothSourceAlpha	Gets a value indicating that the driver supports the Blend.BothSourceAlpha blend mode.
	SupportsDestinationAlpha	Gets a value indicating that the driver supports the Blend.DestinationAlpha blend mode.
	SupportsDestinationColor	Gets a value indicating that the driver supports the Blend.DestinationColor blend mode.
	SupportsInverseDestinationAlpha	Gets a value indicating that the driver supports the Blend.InverseDestinationAlpha blend mode.
	SupportsInverseDestinationColor	Gets a value indicating that the driver supports the Blend.InverseDestinationColor blend mode.
	SupportsInverseSourceAlpha	Gets a value indicating that the driver supports the Blend.InverseSourceAlpha blend mode.
	SupportsInverseSourceColor	Gets a value indicating that the driver supports the Blend.InverseSourceColor blend mode.
	SupportsOne	Gets a value indicating that the driver supports the Blend.One blend mode.
	SupportsSourceAlpha	Gets a value indicating that the driver supports the Blend.SourceAlpha blend mode.
	SupportsSourceAlphaSat	Gets a value indicating that the driver supports the Blend.SourceAlphaSat blend mode.
	SupportsSourceColor	Gets a value indicating that the driver supports the Blend.SourceColor blend mode.
	SupportsZero	Gets a value indicating that the driver supports the Blend.Zero blend mode.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.BlendCaps.SupportsBlendFactor Property

Gets a value indicating that the driver supports the [Blend.BlendFactor](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsBlendFactor { get; }
```

Property Value

true if the driver supports the [Blend.BlendFactor](#) blend mode; **false** otherwise.

Remarks

Constant color blending factor used by the frame-buffer blender.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.BlendFactor](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsBothInverseSourceAlpha Property

Gets a value indicating that the driver supports the [Blend.BothInverseSourceAlpha](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsBothInverseSourceAlpha { get; }
```

Property Value

true if the driver supports the [Blend.BothInverseSourceAlpha](#) blend mode; **false** otherwise.

Remarks

Source blend factor is $(1 - A_S, 1 - A_S, 1 - A_S, 1 - A_S)$, and destination blend factor is (A_S, A_S, A_S, A_S) ; the destination blend selection is overridden. This blend mode is supported only for the [SourceBlend](#) render state.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps](#) Structure

[GraphicsDeviceCapabilities.BlendCaps](#) Members

[Blend.BothInverseSourceAlpha](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsBothSourceAlpha Property

Gets a value indicating that the driver supports the [Blend.BothSourceAlpha](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsBothSourceAlpha { get; }
```

Property Value

true if the driver supports the [Blend.BothSourceAlpha](#) blend mode; **false** otherwise.

Remarks

This mode is obsolete. For DirectX 6 and later, you can achieve the same effect by setting the source and destination blend factors to [Blend.SourceAlpha](#) and [Blend.InverseSourceAlpha](#) in separate calls.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.BothSourceAlpha](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsDestinationAlpha Property

Gets a value indicating that the driver supports the [Blend.DestinationAlpha](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDestinationAlpha { get; }
```

Property Value

true if the driver supports the [Blend.DestinationAlpha](#) blend mode; **false** otherwise.

Remarks

The [Blend.DestinationAlpha](#) blend factor is (A_d, A_d, A_d, A_d) , where A_d is the destination alpha value.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.DestinationAlpha](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsDestinationColor Property

Gets a value indicating that the driver supports the [Blend.DestinationColor](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDestinationColor { get; }
```

Property Value

true if the driver supports the [Blend.DestinationColor](#) blend mode; **false** otherwise.

Remarks

The [Blend.DestinationColor](#) blend factor is (R_d, G_d, B_d, A_d) , where R, G, B, and A respectively stand for red, green, blue, and alpha destination values.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps](#) Structure

[GraphicsDeviceCapabilities.BlendCaps](#) Members

[Blend.DestinationColor](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsInverseDestinationAlpha Property

Gets a value indicating that the driver supports the [Blend.InverseDestinationAlpha](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsInverseDestinationAlpha { get; }
```

Property Value

true if the driver supports the [Blend.InverseDestinationAlpha](#) blend mode; **false** otherwise.

Remarks

The [Blend.InverseDestinationAlpha](#) blend factor is $(1 - A_d, 1 - A_d, 1 - A_d, 1 - A_d)$, where A_d is the alpha destination value.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.InverseDestinationAlpha](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsInverseDestinationColor Property

Gets a value indicating that the driver supports the [Blend.InverseDestinationColor](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsInverseDestinationColor { get; }
```

Property Value

true if the driver supports the [Blend.InverseDestinationColor](#) blend mode; **false** otherwise.

Remarks

The [Blend.InverseDestinationColor](#) blend factor is $(1 - R_d, 1 - G_d, 1 - B_d, 1 - A_d)$, where R_d , G_d , B_d , and A_d respectively stand for the red, green, blue, and alpha destination values.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.InverseDestinationColor](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsInverseSourceAlpha Property

Gets a value indicating that the driver supports the [Blend.InverseSourceAlpha](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsInverseSourceAlpha { get; }
```

Property Value

true if the driver supports the [Blend.InverseSourceAlpha](#) blend mode; **false** otherwise.

Remarks

The [Blend.InverseSourceAlpha](#) blend factor is $(1 - A_S, 1 - A_S, 1 - A_S, 1 - A_S)$, where A_S is the alpha destination value.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.InverseSourceAlpha](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsInverseSourceColor Property

Gets a value indicating that the driver supports the [Blend.InverseSourceColor](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsInverseSourceColor { get; }
```

Property Value

true if the driver supports the [Blend.InverseSourceColor](#) blend mode; **false** otherwise.

Remarks

The [Blend.InverseSourceColor](#) blend factor is $(1 - R_S, 1 - G_S, 1 - B_S, 1 - A_S)$ where R, G, B, and A respectively stand for the red, green, blue, and alpha destination values.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.InverseSourceColor](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsOne Property

Gets a value indicating that the driver supports the [Blend.One](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsOne { get; }
```

Property Value

true if the driver supports the [Blend.One](#) blend mode; **false** otherwise.

Remarks

The [Blend.One](#) blend factor is (1, 1, 1, 1).

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsSourceAlpha Property

Gets a value indicating that the driver supports the [Blend.SourceAlpha](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSourceAlpha { get; }
```

Property Value

true if the driver supports the [Blend.SourceAlpha](#) blend mode; **false** otherwise.

Remarks

The [Blend.SourceAlpha](#) blend factor is (A_S, A_S, A_S, A_S) , where A_S is the alpha source value.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps](#) Structure

[GraphicsDeviceCapabilities.BlendCaps](#) Members

[Blend.SourceAlpha](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsSourceAlphaSat Property

Gets a value indicating that the driver supports the [Blend.SourceAlphaSat](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSourceAlphaSat { get; }
```

Property Value

true if the driver supports the [Blend.SourceAlphaSat](#) blend mode; **false** otherwise.

Remarks

The [Blend.SourceAlphaSat](#) blend factor is $(f, f, f, 1)$, where $f = \min(A, 1 - A_d)$.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps Structure](#)

[GraphicsDeviceCapabilities.BlendCaps Members](#)

[Blend.SourceAlphaSat](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsSourceColor Property

Gets a value indicating that the driver supports the [Blend.SourceColor](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSourceColor { get; }
```

Property Value

true if the driver supports the [Blend.SourceColor](#) blend mode; **false** otherwise.

Remarks

The [Blend.SourceColor](#) blend factor is (R_S, G_S, B_S, A_S) , where R, G, B, and A respectively stand for the red, green, blue, and alpha source values.

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps](#) Structure

[GraphicsDeviceCapabilities.BlendCaps](#) Members

[Blend.SourceColor](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.BlendCaps.SupportsZero Property

Gets a value indicating that the driver supports the [Blend.Zero](#) blend mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsZero { get; }
```

Property Value

true if the driver supports the [Blend.Zero](#) blend mode; **false** otherwise.

Remarks

The [Blend.Zero](#) blend factor is (0, 0, 0, 0).

See Also

Reference

[GraphicsDeviceCapabilities.BlendCaps](#) Structure

[GraphicsDeviceCapabilities.BlendCaps](#) Members

[Blend.Zero](#)

[DestinationBlendCapabilities](#)

[SourceBlendCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps Structure

Represents comparison capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.CompareCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Members](#)









[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.CompareCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.CompareCaps type.



Public Properties

Name	Description
 SupportsAlways	Gets a value indicating whether always passing the comparison test is supported.
 SupportsEqual	Gets a value indicating whether comparison tests in which the new value equals the current value are supported.
 SupportsGreater	Gets a value indicating whether comparison tests in which the new value is greater than the current value are supported.
 SupportsGreaterEqual	Gets a value indicating whether comparison tests in which the new value is greater than or equal to the current value are supported.
 SupportsLess	Gets a value indicating whether comparison tests in which the new value is less than the current value are supported.
 SupportsLessEqual	Gets a value indicating whether comparison tests in which the new value is less than or equal to the current value are supported.
 SupportsNever	Gets a value indicating whether never passing the comparison test is supported.
 SupportsNotEqual	Gets a value indicating whether comparison tests in which the new value does not equal the current value are supported.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 S op_Equality	Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are equal.
 S op_Inequality	Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.CompareCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CompareCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CompareCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.CompareCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.CompareCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.CompareCaps instances are equal.
GraphicsDeviceCapabilities.CompareCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CompareCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.CompareCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.CompareCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.CompareCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    CompareCaps left,  
    CompareCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.CompareCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.CompareCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.CompareCaps.op_Inequality

Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.CompareCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    CompareCaps left,  
    CompareCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.CompareCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.CompareCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.CompareCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)









[GraphicsDeviceCapabilities.CompareCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps Properties

Public Properties

	Name	Description
	SupportsAlways	Gets a value indicating whether always passing the comparison test is supported.
	SupportsEqual	Gets a value indicating whether comparison tests in which the new value equals the current value are supported.
	SupportsGreater	Gets a value indicating whether comparison tests in which the new value is greater than the current value are supported.
	SupportsGreaterEqual	Gets a value indicating whether comparison tests in which the new value is greater than or equal to the current value are supported.
	SupportsLess	Gets a value indicating whether comparison tests in which the new value is less than the current value are supported.
	SupportsLessEqual	Gets a value indicating whether comparison tests in which the new value is less than or equal to the current value are supported.
	SupportsNever	Gets a value indicating whether never passing the comparison test is supported.
	SupportsNotEqual	Gets a value indicating whether comparison tests in which the new value does not equal the current value are supported.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CompareCaps.SupportsAlways Property

Gets a value indicating whether always passing the comparison test is supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAlways { get; }
```

Property Value

true if always passing the comparison test is supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsEqual Property

Gets a value indicating whether comparison tests in which the new value equals the current value are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsEqual { get; }
```

Property Value

true if comparison tests in which the new value equals the current value are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsGreater Property

Gets a value indicating whether comparison tests in which the new value is greater than the current value are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsGreater { get; }
```

Property Value

true if comparison tests in which the new value is greater than the current value are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsGreaterEqual Property

Gets a value indicating whether comparison tests in which the new value is greater than or equal to the current value are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsGreaterEqual { get; }
```

Property Value

true if comparison tests in which the new value is greater than or equal to the current value are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsLess Property

Gets a value indicating whether comparison tests in which the new value is less than the current value are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsLess { get; }
```

Property Value

true if comparison tests in which the new value is less than the current value are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsLessEqual Property

Gets a value indicating whether comparison tests in which the new value is less than or equal to the current value are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsLessEqual { get; }
```

Property Value

true if comparison tests in which the new value is less than or equal to the current value are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsNever Property

Gets a value indicating whether never passing the comparison test is supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNever { get; }
```

Property Value

true if never passing the comparison test is supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CompareCaps.SupportsNotEqual Property

Gets a value indicating whether comparison tests in which the new value does not equal the current value are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNotEqual { get; }
```

Property Value

true if comparison tests in which the new value does not equal the current value are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CompareCaps Structure](#)

[GraphicsDeviceCapabilities.CompareCaps Members](#)

[AlphaCompareCapabilities](#)

[DepthBufferCompareCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CursorCaps Structure

Represents hardware support for cursors.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.CursorCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










GraphicsDeviceCapabilities.CursorCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.CursorCaps type.



Public Properties

	Name	Description
	SupportsColor	Gets a value indicating whether a full-color cursor is supported in hardware in high-resolution modes.
	SupportsLowResolution	Gets a value indicating whether a full-color cursor is supported in hardware in low-resolution modes.

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
 	Op_Equality	Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are equal.
 	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.CursorCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CursorCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CursorCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.CursorCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.CursorCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.CursorCaps instances are equal.
GraphicsDeviceCapabilities.CursorCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CursorCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.CursorCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.CursorCaps](#) to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CursorCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CursorCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.CursorCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    CursorCaps left,  
    CursorCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.CursorCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.CursorCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.CursorCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.CursorCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    CursorCaps left,  
    CursorCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.CursorCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.CursorCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.CursorCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)



[GraphicsDeviceCapabilities.CursorCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CursorCaps Properties

Public Properties

	Name	Description
	SupportsColor	Gets a value indicating whether a full-color cursor is supported in hardware in high-resolution modes.
	SupportsLowResolution	Gets a value indicating whether a full-color cursor is supported in hardware in low-resolution modes.

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[CursorCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.CursorCaps.SupportsColor Property

Gets a value indicating whether a full-color cursor is supported in hardware in high-resolution modes.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsColor { get; }
```

Property Value

true if the hardware supports a full-color cursor in high resolution; **false** otherwise.

Remarks

Specifically, this flag indicates that the driver supports at least a hardware color cursor in high-resolution modes (with scan lines greater than or equal to 400).

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[CursorCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.CursorCaps.SupportsLowResolution Property

Gets a value indicating whether a full-color cursor is supported in hardware in low-resolution modes.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsLowResolution { get; }
```

Property Value

true if the hardware supports a full-color cursor in low resolution; **false** otherwise.

Remarks

Specifically, this flag indicates that the driver supports at least a hardware color cursor in low-resolution modes (with scan lines less than 400).

See Also

Reference

[GraphicsDeviceCapabilities.CursorCaps Structure](#)

[GraphicsDeviceCapabilities.CursorCaps Members](#)

[CursorCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps Structure

Represents data types contained in a vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.DeclarationTypeCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[DeclarationTypeCapabilities](#)











[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.DeclarationTypeCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.DeclarationTypeCaps type.



Public Properties

	Name	Description
	SupportsByte4	Gets a value indicating whether vertex declarations support VertexElementFormat.Byte4 .
	SupportsHalfVector2	Gets a value indicating whether vertex declarations support VertexElementFormat.HalfVector2 .
	SupportsHalfVector4	Gets a value indicating whether vertex declarations support VertexElementFormat.HalfVector4 .
	SupportsNormalized101010	Gets a value indicating whether vertex declarations support VertexElementFormat.Normalized101010 .
	SupportsNormalizedShort2	Gets a value indicating whether vertex declarations support VertexElementFormat.NormalizedShort2 .
	SupportsNormalizedShort4	Gets a value indicating whether vertex declarations support VertexElementFormat.NormalizedShort4 .
	SupportsRg32	Gets a value indicating whether vertex declarations support VertexElementFormat.Rg32 .
	SupportsRgba32	Gets a value indicating whether vertex declarations support VertexElementFormat.Rgba32 .
	SupportsRgba64	Gets a value indicating whether vertex declarations support VertexElementFormat.Rgba64 .
	SupportsUInt101010	Gets a value indicating whether vertex declarations support VertexElementFormat.UInt101010 .

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also










Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeclarationTypeCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are equal.
 	op_Inequality	Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeclarationTypeCaps.Equals

Method

Determines whether two [GraphicsDeviceCapabilities.DeclarationTypeCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.DeclarationTypeCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.DeclarationTypeCaps instances are equal.
GraphicsDeviceCapabilities.DeclarationTypeCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeclarationTypeCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.DeclarationTypeCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.DeclarationTypeCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.DeclarationTypeCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    DeclarationTypeCaps left,  
    DeclarationTypeCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.DeclarationTypeCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.DeclarationTypeCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    DeclarationTypeCaps left,  
    DeclarationTypeCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDeviceCapabilities.DeclarationTypeCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)











[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps Properties

Public Properties

	Name	Description
	SupportsByte4	Gets a value indicating whether vertex declarations support VertexElementFormat.Byte4 .
	SupportsHalfVector2	Gets a value indicating whether vertex declarations support VertexElementFormat.HalfVector2 .
	SupportsHalfVector4	Gets a value indicating whether vertex declarations support VertexElementFormat.HalfVector4 .
	SupportsNormalized101010	Gets a value indicating whether vertex declarations support VertexElementFormat.Normalized101010 .
	SupportsNormalizedShort2	Gets a value indicating whether vertex declarations support VertexElementFormat.NormalizedShort2 .
	SupportsNormalizedShort4	Gets a value indicating whether vertex declarations support VertexElementFormat.NormalizedShort4 .
	SupportsRg32	Gets a value indicating whether vertex declarations support VertexElementFormat.Rg32 .
	SupportsRgba32	Gets a value indicating whether vertex declarations support VertexElementFormat.Rgba32 .
	SupportsRgba64	Gets a value indicating whether vertex declarations support VertexElementFormat.Rgba64 .
	SupportsUInt101010	Gets a value indicating whether vertex declarations support VertexElementFormat.UInt101010 .

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsByte4 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.Byte4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsByte4 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.Byte4](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsHalfVector2 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.HalfVector2](#)

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsHalfVector2 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.HalfVector2](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsHalfVector4 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.HalfVector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsHalfVector4 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.HalfVector4](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsNormalized101010 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.Normalized101010](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNormalized101010 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.Normalized101010](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsNormalizedShort2 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.NormalizedShort2](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNormalizedShort2 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.NormalizedShort2](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsNormalizedShort4 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.NormalizedShort4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNormalizedShort4 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.NormalizedShort4](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsRg32 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.Rg32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsRg32 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.Rg32](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps Structure](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Members](#)

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsRgba32 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.Rgba32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsRgba32 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.Rgba32](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsRgba64 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.Rgba64](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsRgba64 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.Rgba64](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeclarationTypeCaps.SupportsUInt101010 Property

Gets a value indicating whether vertex declarations support [VertexElementFormat.UInt101010](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsUInt101010 { get; }
```

Property Value

true if vertex declarations support [VertexElementFormat.UInt101010](#); otherwise false.

See Also

Reference

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Structure

[GraphicsDeviceCapabilities.DeclarationTypeCaps](#) Members

[DeclarationTypeCapabilities](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps Structure

Represents device-specific capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.DeviceCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







GraphicsDeviceCapabilities.DeviceCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.DeviceCaps type.



Public Properties

Name	Description
 CanDrawSystemToNonLocal	Gets a value indicating whether the device supports blits from system-memory textures to non-local video-memory textures.
 CanRenderAfterFlip	Gets a value indicating whether the device can queue rendering commands after a page flip.
 IsDirect3D9Driver	Gets a value indicating if the device supports copying the contents of a source rectangle to a destination rectangle using a texture as the source.
 SupportsDrawPrimitives2	Gets a value indicating whether the device can support at least a DirectX 5-compliant driver.
 SupportsDrawPrimitives2Ex	Gets a value indicating whether the device can support at least a DirectX 7-compliant driver.
 SupportsDrawPrimitivesTransformedVertex	Gets a value indicating whether the device exports a DrawPrimitives -aware hardware abstraction layer (HAL).
 SupportsExecuteSystemMemory	Gets a value indicating whether the device can use execute buffers from video memory.
 SupportsExecuteVideoMemory	Gets a value indicating whether the device can use execute buffers from video memory.
 SupportsHardwareRasterization	Gets a value indicating whether the device has hardware acceleration for scene rasterization.
 SupportsHardwareTransformAndLight	Gets a value indicating whether the device can support transformation and lighting in hardware.
 SupportsSeparateTextureMemories	Gets a value indicating whether the device is texturing from separate memory pools.
 SupportsStreamOffset	Gets a value indicating whether the device supports stream offsets.
 SupportsTextureNonLocalVideoMemory	Gets a value indicating whether the device can retrieve textures from non-local video memory.
 SupportsTextureSystemMemory	Gets a value indicating whether the device can retrieve textures from system memory.
 SupportsTextureVideoMemory	Gets a value indicating whether the device can retrieve textures from device memory.
 SupportsTransformedVertexSystemMemory	Gets a value indicating whether the device can use buffers from system memory for transformed and lit vertices.
 SupportsTransformedVertexVideoMemory	Gets a value indicating whether the device can use buffers from video memory for transformed and lit vertices.
 VertexElementScanSharesStreamOffset	Gets a value indicating whether the device allows multiple vertex elements to share the same offset in a stream.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are equal.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)







See Also

Reference



[GraphicsDeviceCapabilities.DeviceCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeviceCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are equal.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeviceCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.DeviceCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.DeviceCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.DeviceCaps instances are equal.
GraphicsDeviceCapabilities.DeviceCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeviceCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.DeviceCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.DeviceCaps](#) to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.DeviceCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    DeviceCaps left,  
    DeviceCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.DeviceCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.DeviceCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.DeviceCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.DeviceCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    DeviceCaps left,  
    DeviceCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.DeviceCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.DeviceCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.DeviceCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps Properties

Public Properties

Name	Description
 CanDrawSystemToNonLocal	Gets a value indicating whether the device supports blits from system-memory textures to non-local video-memory textures.
 CanRenderAfterFlip	Gets a value indicating whether the device can queue rendering commands after a page flip.
 IsDirect3D9Driver	Gets a value indicating if the device supports copying the contents of a source rectangle to a destination rectangle using a texture as the source.
 SupportsDrawPrimitives2	Gets a value indicating whether the device can support at least a DirectX 5-compliant driver.
 SupportsDrawPrimitives2Ex	Gets a value indicating whether the device can support at least a DirectX 7-compliant driver.
 SupportsDrawPrimitivesTransformedVertex	Gets a value indicating whether the device exports a DrawPrimitives -aware hardware abstraction layer (HAL).
 SupportsExecuteSystemMemory	Gets a value indicating whether the device can use execute buffers from video memory.
 SupportsExecuteVideoMemory	Gets a value indicating whether the device can use execute buffers from video memory.
 SupportsHardwareRasterization	Gets a value indicating whether the device has hardware acceleration for scene rasterization.
 SupportsHardwareTransformAndLight	Gets a value indicating whether the device can support transformation and lighting in hardware.
 SupportsSeparateTextureMemories	Gets a value indicating whether the device is texturing from separate memory pools.
 SupportsStreamOffset	Gets a value indicating whether the device supports stream offsets.
 SupportsTextureNonLocalVideoMemory	Gets a value indicating whether the device can retrieve textures from non-local video memory.
 SupportsTextureSystemMemory	Gets a value indicating whether the device can retrieve textures from system memory.
 SupportsTextureVideoMemory	Gets a value indicating whether the device can retrieve textures from device memory.
 SupportsTransformedVertexSystemMemory	Gets a value indicating whether the device can use buffers from system memory for transformed and lit vertices.
 SupportsTransformedVertexVideoMemory	Gets a value indicating whether the device can use buffers from video memory for transformed and lit vertices.
 VertexElementScanSharesStreamOffset	Gets a value indicating whether the device allows multiple vertex elements to share the same offset in a stream.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DeviceCaps.CanDrawSystemToNonLocal Property

Gets a value indicating whether the device supports blits from system-memory textures to non-local video-memory textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CanDrawSystemToNonLocal { get; }
```

Property Value

true if the device supports blits from system-memory textures to non-local video-memory textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.CanRenderAfterFlip Property

Gets a value indicating whether the device can queue rendering commands after a page flip.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CanRenderAfterFlip { get; }
```

Property Value

true if the device can queue rendering commands after a page flip; **false** otherwise.

Remarks

Applications do not change their behavior if this flag is set; this capability means that the device is relatively fast.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.IsDirect3D9Driver Property

Gets a value indicating if the device supports copying the contents of a source rectangle to a destination rectangle using a texture as the source.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDirect3D9Driver { get; }
```

Property Value

true if the device supports copying the contents of a source rectangle to a destination rectangle using a texture as the source; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsDrawPrimitives2 Property

Gets a value indicating whether the device can support at least a DirectX 5–compliant driver.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDrawPrimitives2 { get; }
```

Property Value

true if the device can support at least a DirectX 5–compliant driver; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsDrawPrimitives2Ex Property

Gets a value indicating whether the device can support at least a DirectX 7–compliant driver.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDrawPrimitives2Ex { get; }
```

Property Value

true if the device can support at least a DirectX 7–compliant driver; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsDrawPrimitivesTransformedVertex Property

Gets a value indicating whether the device exports a [DrawPrimitives](#)-aware hardware abstraction layer (HAL).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDrawPrimitivesTransformedVertex { get; }
```

Property Value

true if the device exports an [DrawPrimitives](#)-aware HAL; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsExecuteSystemMemory Property

Gets a value indicating whether the device can use execute buffers from video memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsExecuteSystemMemory { get; }
```

Property Value

true if the device can use execute buffers from video memory; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsExecuteVideoMemory Property

Gets a value indicating whether the device can use execute buffers from video memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsExecuteVideoMemory { get; }
```

Property Value

true if the device can use execute buffers from video memory; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsHardwareRasterization Property

Gets a value indicating whether the device has hardware acceleration for scene rasterization.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsHardwareRasterization { get; }
```

Property Value

true if the device has hardware acceleration for scene rasterization; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsHardwareTransformAndLight Property

Gets a value indicating whether the device can support transformation and lighting in hardware.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsHardwareTransformAndLight { get; }
```

Property Value

true if the device can support transformation and lighting in hardware; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsSeparateTextureMemories Property

Gets a value indicating whether the device is texturing from separate memory pools.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSeparateTextureMemories { get; }
```

Property Value

true if the device is texturing from separate memory pools; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsStreamOffset Property

Gets a value indicating whether the device supports stream offsets.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsStreamOffset { get; }
```

Property Value

true if the device supports stream offsets; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsTextureNonLocalVideoMemory Property

Gets a value indicating whether the device can retrieve textures from non-local video memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureNonLocalVideoMemory { get; }
```

Property Value

true if the device can retrieve textures from non-local video memory; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsTextureSystemMemory Property

Gets a value indicating whether the device can retrieve textures from system memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureSystemMemory { get; }
```

Property Value

true if the device can retrieve textures from system memory; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsTextureVideoMemory Property

Gets a value indicating whether the device can retrieve textures from device memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureVideoMemory { get; }
```

Property Value

true if the device can retrieve textures from device memory; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsTransformedVertexSystemMemory Property

Gets a value indicating whether the device can use buffers from system memory for transformed and lit vertices.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTransformedVertexSystemMemory { get; }
```

Property Value

true if the device can use buffers from system memory for transformed and lit vertices; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.SupportsTransformedVertexVideoMemory Property

Gets a value indicating whether the device can use buffers from video memory for transformed and lit vertices.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTransformedVertexVideoMemory { get; }
```

Property Value

true if the device can use buffers from video memory for transformed and lit vertices; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DeviceCaps.VertexElementScanSharesStreamOffset Property

Gets a value indicating whether the device allows multiple vertex elements to share the same offset in a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool VertexElementScanSharesStreamOffset { get; }
```

Property Value

true if the device allows multiple vertex elements to share the same offset in a stream; **false** otherwise.

Remarks

Multiple vertex elements can share the same offset in a stream if **VertexElementScanSharesStreamOffset** is **true** and the vertex declaration does not have an element with [VertexElementUsage.PositionTransformed](#).

See Also

Reference

[GraphicsDeviceCapabilities.DeviceCaps Structure](#)

[GraphicsDeviceCapabilities.DeviceCaps Members](#)

[DeviceCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps Structure

Represents driver-specific capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.DriverCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)











[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.DriverCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.DriverCaps type.



Public Properties

Name	Description
 CanAutoGenerateMipMap	Gets a value indicating whether the driver is capable of automatically generating mipmaps.
 CanCalibrateGamma	Gets a value indicating whether the system has a calibrator installed that can automatically adjust the gamma ramp.
 CanManageResource	Gets a value indicating whether the driver is capable of managing resources.
 ReadScanLine	Gets a value indicating whether the display hardware is capable of returning the current scan line.
 SupportsAlphaFullScreenFlipOrDiscard	Gets a value indicating whether the device can respect the AlphaBlendEnable render state in full-screen mode while using the FLIP or DISCARD swap effect.
 SupportsCopyToSystemMemory	Gets a value indicating whether the device can accelerate a memory copy from local video memory to system memory.
 SupportsCopyToVideoMemory	Gets a value indicating whether the device can accelerate a memory copy from system memory to local video memory.
 SupportsDynamicTextures	Gets a value indicating whether the driver supports dynamic textures.
 SupportsFullScreenGamma	Gets a value indicating whether the driver supports dynamic gamma ramp adjustment in full-screen mode.
 SupportsLinearToSrgbPresentation	Gets a value indicating whether the device can perform gamma correction from a windowed back buffer (containing linear content) to an sRGB desktop.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are equal.
 GetHashCode	gets the hash code for this object.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.DriverCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DriverCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are equal.
	GetHashCode	gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DriverCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.DriverCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.DriverCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.DriverCaps instances are equal.
GraphicsDeviceCapabilities.DriverCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DriverCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.DriverCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.DriverCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.GetHashCode Method

gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.DriverCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    DriverCaps left,  
    DriverCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.DriverCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.DriverCaps](#) on the left side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.DriverCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.DriverCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    DriverCaps left,  
    DriverCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.DriverCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.DriverCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.DriverCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)











[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps Properties

Public Properties

	Name	Description
	CanAutoGenerateMipMap	Gets a value indicating whether the driver is capable of automatically generating mipmaps.
	CanCalibrateGamma	Gets a value indicating whether the system has a calibrator installed that can automatically adjust the gamma ramp.
	CanManageResource	Gets a value indicating whether the driver is capable of managing resources.
	ReadScanLine	Gets a value indicating whether the display hardware is capable of returning the current scan line.
	SupportsAlphaFullscreenFlipOrDiscard	Gets a value indicating whether the device can respect the AlphaBlendEnable render state in full-screen mode while using the FLIP or DISCARD swap effect.
	SupportsCopyToSystemMemory	Gets a value indicating whether the device can accelerate a memory copy from local video memory to system memory.
	SupportsCopyToVideoMemory	Gets a value indicating whether the device can accelerate a memory copy from system memory to local video memory.
	SupportsDynamicTextures	Gets a value indicating whether the driver supports dynamic textures.
	SupportsFullscreenGamma	Gets a value indicating whether the driver supports dynamic gamma ramp adjustment in full-screen mode.
	SupportsLinearToSrgbPresentation	Gets a value indicating whether the device can perform gamma correction from a windowed back buffer (containing linear content) to an sRGB desktop.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.DriverCaps.CanAutoGenerateMipMap Property

Gets a value indicating whether the driver is capable of automatically generating mipmaps.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CanAutoGenerateMipMap { get; }
```

Property Value

true if the driver is capable of automatically generating mipmaps; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.CanCalibrateGamma Property

Gets a value indicating whether the system has a calibrator installed that can automatically adjust the gamma ramp.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CanCalibrateGamma { get; }
```

Property Value

true if the system has a calibrator installed that can automatically adjust the gamma ramp; **false** otherwise.

Remarks

To invoke the calibrator when setting new gamma levels, set *calibrate* to **true** when calling [SetGammaRamp](#). Calibrating gamma ramps incurs some processing overhead and should not be used frequently.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.CanManageResource Property

Gets a value indicating whether the driver is capable of managing resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool CanManageResource { get; }
```

Property Value

true if the driver is capable of managing resources; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.ReadScanLine Property

Gets a value indicating whether the display hardware is capable of returning the current scan line.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool ReadScanLine { get; }
```

Property Value

true if the driver display hardware is capable of returning the current scan line; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.SupportsAlphaFullScreenFlipOrDiscard Property

Gets a value indicating whether the device can respect the [AlphaBlendEnable](#) render state in full-screen mode while using the FLIP or DISCARD swap effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAlphaFullScreenFlipOrDiscard { get; }
```

Property Value

true if the driver the device can respect the [AlphaBlendEnable](#) render state in full-screen mode while using the FLIP or DISCARD swap effect; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.SupportsCopyToSystemMemory Property

Gets a value indicating whether the device can accelerate a memory copy from local video memory to system memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsCopyToSystemMemory { get; }
```

Property Value

true if the device can accelerate a memory copy from local video memory to system memory; **false** otherwise.

Remarks

This setting guarantees that [GetRenderTarget](#) calls will be hardware accelerated. If this setting is **false**, those calls will succeed but will be slower.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.SupportsCopyToVideoMemory Property

Gets a value indicating whether the device can accelerate a memory copy from system memory to local video memory.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsCopyToVideoMemory { get; }
```

Property Value

true if the device can accelerate a memory copy from system memory to local video memory; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.SupportsDynamicTextures Property

Gets a value indicating whether the driver supports dynamic textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDynamicTextures { get; }
```

Property Value

true if the driver supports dynamic textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.SupportsFullScreenGamma Property

Gets a value indicating whether the driver supports dynamic gamma ramp adjustment in full-screen mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFullScreenGamma { get; }
```

Property Value

true if the driver supports dynamic gamma ramp adjustment in full-screen mode; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.DriverCaps.SupportsLinearToSrgbPresentation Property

Gets a value indicating whether the device can perform gamma correction from a windowed back buffer (containing linear content) to an sRGB desktop.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsLinearToSrgbPresentation { get; }
```

Property Value

true if the device can perform gamma correction from a windowed back buffer to an sRGB desktop; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.DriverCaps Structure](#)

[GraphicsDeviceCapabilities.DriverCaps Members](#)

[DriverCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps Structure

Represents texture filter capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.FilterCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)













[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.FilterCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.FilterCaps type.



Public Properties

Name	Description
 SupportsMagnifyAnisotropic	Gets a value that indicates that the device supports per-stage anisotropic filtering for magnifying textures.
 SupportsMagnifyGaussianQuad	Gets a value indicating whether the device supports per-stage Gaussian quad filtering for magnifying textures.
 SupportsMagnifyLinear	Gets a value that indicates that the device supports per-stage bilinear interpolation filtering for magnifying textures.
 SupportsMagnifyPoint	Gets a value that indicates that the device supports per-stage point-sample filtering for magnifying textures.
 SupportsMagnifyPyramidalQuad	Gets a value that indicates that the device supports per-stage pyramidal sample filtering for magnifying textures.
 SupportsMinifyAnisotropic	Gets a value that indicates that the device supports per-stage anisotropic filtering for minifying textures.
 SupportsMinifyGaussianQuad	Gets a value indicating whether the device supports per-stage Gaussian quad filtering for minifying textures.
 SupportsMinifyLinear	Gets a value that indicates that the device supports per-stage bilinear interpolation filtering for minifying textures.
 SupportsMinifyPoint	Gets a value that indicates whether the device supports per-stage point-sample filtering for minifying textures.
 SupportsMinifyPyramidalQuad	Gets a value that indicates that the device supports per-stage pyramidal sample filtering for minifying textures.
 SupportsMipMapLinear	Gets a value that indicates that the device supports per-stage trilinear interpolation filtering for mipmaps.
 SupportsMipMapPoint	Gets a value that indicates that the device supports per-stage point-sample filtering for mipmaps.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 S op_Equality	Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are equal.
 S op_Inequality	Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.FilterCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.FilterCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.FilterCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.FilterCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.FilterCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.FilterCaps instances are equal.
GraphicsDeviceCapabilities.FilterCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.FilterCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.FilterCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.FilterCaps](#) to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.FilterCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    FilterCaps left,  
    FilterCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.FilterCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.FilterCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.FilterCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.FilterCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    FilterCaps left,  
    FilterCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.FilterCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.FilterCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.FilterCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)













[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps Properties

Public Properties

	Name	Description
	SupportsMagnifyAnisotropic	Gets a value that indicates that the device supports per-stage anisotropic filtering for magnifying textures.
	SupportsMagnifyGaussianQuad	Gets a value indicating whether the device supports per-stage Gaussian quad filtering for magnifying textures.
	SupportsMagnifyLinear	Gets a value that indicates that the device supports per-stage bilinear interpolation filtering for magnifying textures.
	SupportsMagnifyPoint	Gets a value that indicates that the device supports per-stage point-sample filtering for magnifying textures.
	SupportsMagnifyPyramidalQuad	Gets a value that indicates that the device supports per-stage pyramidal sample filtering for magnifying textures.
	SupportsMinifyAnisotropic	Gets a value that indicates that the device supports per-stage anisotropic filtering for minifying textures.
	SupportsMinifyGaussianQuad	Gets a value indicating whether the device supports per-stage Gaussian quad filtering for minifying textures.
	SupportsMinifyLinear	Gets a value that indicates that the device supports per-stage bilinear interpolation filtering for minifying textures.
	SupportsMinifyPoint	Gets a value that indicates whether the device supports per-stage point-sample filtering for minifying textures.
	SupportsMinifyPyramidalQuad	Gets a value that indicates that the device supports per-stage pyramidal sample filtering for minifying textures.
	SupportsMipMapLinear	Gets a value that indicates that the device supports per-stage trilinear interpolation filtering for mipmaps.
	SupportsMipMapPoint	Gets a value that indicates that the device supports per-stage point-sample filtering for mipmaps.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.FilterCaps.SupportsMagnifyAnisotropic Property

Gets a value that indicates that the device supports per-stage anisotropic filtering for magnifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMagnifyAnisotropic { get; }
```

Property Value

true if the device supports per-stage anisotropic filtering for magnifying textures; **false** otherwise.

Remarks

Anisotropic filters compensate for distortion caused by the difference in angle between the texture polygon and the plane of the screen.

The anisotropic magnification filter is represented by the [TextureFilter.Anisotropic](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMagnifyGaussianQuad Property

Gets a value indicating whether the device supports per-stage Gaussian quad filtering for magnifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMagnifyGaussianQuad { get; }
```

Property Value

true if the device supports per-stage Gaussian quad filtering for magnifying textures; **false** otherwise.

Remarks

The Gaussian quad filter is represented by the [TextureFilter.GaussianQuad](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMagnifyLinear Property

Gets a value that indicates that the device supports per-stage bilinear interpolation filtering for magnifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMagnifyLinear { get; }
```

Property Value

true if the device supports per-stage bilinear interpolation filtering for magnifying textures; **false** otherwise.

Remarks

The bilinear interpolation mipmapping filter is represented by the [TextureFilter.Linear](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMagnifyPoint Property

Gets a value that indicates that the device supports per-stage point-sample filtering for magnifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMagnifyPoint { get; }
```

Property Value

true if the device supports per-stage point-sample filtering for magnifying textures; **false** otherwise.

Remarks

The point-sample magnification filter is represented by the [TextureFilter.Point](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMagnifyPyramidalQuad Property

Gets a value that indicates that the device supports per-stage pyramidal sample filtering for magnifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMagnifyPyramidalQuad { get; }
```

Property Value

true if the device supports per-stage pyramidal sample filtering for magnifying textures; **false** otherwise.

Remarks

The pyramidal sample filter is represented by the [TextureFilter.PyramidalQuad](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMinifyAnisotropic Property

Gets a value that indicates that the device supports per-stage anisotropic filtering for minifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMinifyAnisotropic { get; }
```

Property Value

true if the device supports per-stage anisotropic filtering for minifying textures; **false** otherwise.

Remarks

The anisotropic minification filter is represented by the [TextureFilter.Anisotropic](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMinifyGaussianQuad Property

Gets a value indicating whether the device supports per-stage Gaussian quad filtering for minifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMinifyGaussianQuad { get; }
```

Property Value

true if the device supports per-stage Gaussian quad filtering for magnifying textures; **false** otherwise.

Remarks

The Gaussian quad filter is represented by the [TextureFilter.GaussianQuad](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMinifyLinear Property

Gets a value that indicates that the device supports per-stage bilinear interpolation filtering for minifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMinifyLinear { get; }
```

Property Value

true if the device supports per-stage bilinear interpolation filtering for minifying textures; **false** otherwise.

Remarks

The linear minification filter is represented by the [TextureFilter.Linear](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMinifyPoint Property

Gets a value that indicates whether the device supports per-stage point-sample filtering for minifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMinifyPoint { get; }
```

Property Value

true if the device supports per-stage point-sample filtering for minifying textures; **false** otherwise.

Remarks

The point-sample minification filter is represented by the [TextureFilter.Point](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMinifyPyramidalQuad Property

Gets a value that indicates that the device supports per-stage pyramidal sample filtering for minifying textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMinifyPyramidalQuad { get; }
```

Property Value

true if the device supports per-stage pyramidal sample filtering for minifying textures; **false** otherwise.

Remarks

The pyramidal sample filter is represented by the [TextureFilter.PyramidalQuad](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMipMapLinear Property

Gets a value that indicates that the device supports per-stage trilinear interpolation filtering for mipmaps.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMipMapLinear { get; }
```

Property Value

true if the device supports per-stage trilinear interpolation filtering for mipmaps; **false** otherwise.

Remarks

The bilinear interpolation mipmapping filter is represented by the [TextureFilter.Linear](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.FilterCaps.SupportsMipMapPoint Property

Gets a value that indicates that the device supports per-stage point-sample filtering for mipmaps.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMipMapPoint { get; }
```

Property Value

true if the device supports per-stage point-sample filtering for mipmaps; **false** otherwise.

Remarks

The point-sample mipmapping filter is represented by the [TextureFilter.Point](#) enumerated value.

See Also

Reference

[GraphicsDeviceCapabilities.FilterCaps Structure](#)

[GraphicsDeviceCapabilities.FilterCaps Members](#)

[CubeTextureFilterCapabilities](#)

[TextureFilterCapabilities](#)

[VertexTextureFilterCapabilities](#)

[VolumeTextureFilterCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps Structure

Represents the capabilities for line-drawing primitives.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.LineCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)







[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.LineCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.LineCaps type.



Public Properties

	Name	Description
	SupportsAlphaCompare	Gets a value indicating whether the device supports alpha-test comparisons.
	SupportsAntiAlias	Gets a value indicating whether the device supports antialiased lines.
	SupportsBlend	Gets a value indicating whether the device supports source blending.
	SupportsDepthBufferTest	Gets a value indicating whether the device supports depth buffer comparisons.
	SupportsFog	Gets a value indicating whether the device supports fog.
	SupportsTextureMapping	Gets a value indicating whether the device supports texture mapping.

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.LineCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.LineCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.LineCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.LineCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.LineCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.LineCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.LineCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.LineCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.LineCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.LineCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.LineCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.LineCaps instances are equal.
GraphicsDeviceCapabilities.LineCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.LineCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.LineCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.LineCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.LineCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    LineCaps left,  
    LineCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.LineCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.LineCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.LineCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.LineCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    LineCaps left,  
    LineCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.LineCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.LineCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.LineCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)







[GraphicsDeviceCapabilities.LineCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps Properties

Public Properties

	Name	Description
	SupportsAlphaCompare	Gets a value indicating whether the device supports alpha-test comparisons.
	SupportsAntiAlias	Gets a value indicating whether the device supports antialiased lines.
	SupportsBlend	Gets a value indicating whether the device supports source blending.
	SupportsDepthBufferTest	Gets a value indicating whether the device supports depth buffer comparisons.
	SupportsFog	Gets a value indicating whether the device supports fog.
	SupportsTextureMapping	Gets a value indicating whether the device supports texture mapping.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.LineCaps.SupportsAlphaCompare Property

Gets a value indicating whether the device supports alpha-test comparisons.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAlphaCompare { get; }
```

Property Value

true if the device supports alpha-test comparisons; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.SupportsAntiAlias Property

Gets a value indicating whether the device supports antialiased lines.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAntiAlias { get; }
```

Property Value

true if the device supports antialiased lines; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.SupportsBlend Property

Gets a value indicating whether the device supports source blending.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsBlend { get; }
```

Property Value

true if the device supports source blending; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.SupportsDepthBufferTest Property

Gets a value indicating whether the device supports depth buffer comparisons.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDepthBufferTest { get; }
```

Property Value

true if the device supports depth buffer comparisons; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.SupportsFog Property

Gets a value indicating whether the device supports fog.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFog { get; }
```

Property Value

true if the device supports fog; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.LineCaps.SupportsTextureMapping Property

Gets a value indicating whether the device supports texture mapping.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureMapping { get; }
```

Property Value

true if the device supports texture mapping; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.LineCaps Structure](#)

[GraphicsDeviceCapabilities.LineCaps Members](#)

[LineCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps Structure

Represents pixel shader capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.PixelShaderCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)









[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










GraphicsDeviceCapabilities.PixelShaderCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.PixelShaderCaps type.








Public Fields

Name	Description
 MaxDynamicFlowControlDepth	Specifies the maximum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if_pred - vs).
 MaxNumberInstructionSlots	Specifies the maximum number of instruction slots supported.
 MaxNumberTemps	Specifies the maximum number of temporary registers supported.
 MaxStaticFlowControlDepth	Specifies the depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions
 MinDynamicFlowControlDepth	Specifies the minimum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if_pred - vs).
 MinNumberInstructionSlots	Specifies the minimum number of instruction slots supported.
 MinNumberTemps	Specifies the minimum number of temporary registers supported.
 MinStaticFlowControlDepth	Specifies the minimum depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.



Public Properties

Name	Description
 DynamicFlowControlDepth	Gets a value specifying the dynamic flow control depth.
 NumberInstructionSlots	Gets a value specifying the number of instruction slots supported.
 NumberTemps	Gets a value specifying the number of temporary registers supported.
 StaticFlowControlDepth	Gets a value specifying the depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.
 SupportsArbitrarySwizzle	Gets a value indicating whether arbitrary swizzling is supported.
 SupportsGradientInstructions	Gets a value indicating whether gradient instructions are supported.
 SupportsNoDependentReadLimit	Gets a value indicating whether there is a limit on the number of dependent reads per instruction.
 SupportsNoTextureInstructionLimit	Gets a value indicating whether there is a limit on the number of texture instructions.
 SupportsPredication	Gets a value indicating whether instruction predication is supported.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PixelShaderCaps Fields

Public Fields

Name	Description
MaxDynamicFlowControlDepth	Specifies the maximum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if_pred - vs).
MaxNumberInstructionSlots	Specifies the maximum number of instruction slots supported.
MaxNumberTemps	Specifies the maximum number of temporary registers supported.
MaxStaticFlowControlDepth	Specifies the depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions
MinDynamicFlowControlDepth	Specifies the minimum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if_pred - vs).
MinNumberInstructionSlots	Specifies the minimum number of instruction slots supported.
MinNumberTemps	Specifies the minimum number of temporary registers supported.
MinStaticFlowControlDepth	Specifies the minimum depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PixelShaderCaps.MaxDynamicFlowControlDepth Field

Specifies the maximum level of nesting of dynamic flow control instructions (**break - vs**, **break_comp - vs**, **breakp - vs**, **if_comp - vs**, **if_comp - vs**, **if_pred - vs**).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxDynamicFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MaxNumberInstructionSlots Field

Specifies the maximum number of instruction slots supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxNumberInstructionSlots
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MaxNumberTemps Field

Specifies the maximum number of temporary registers supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxNumberTemps
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MaxStaticFlowControlDepth Field

Specifies the depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxStaticFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MinDynamicFlowControlDepth Field

Specifies the minimum level of nesting of dynamic flow control instructions (**break - vs**, **break_comp - vs**, **breakp - vs**, **if_comp - vs**, **if_comp - vs**, **if_pred - vs**).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinDynamicFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MinNumberInstructionSlots Field

Specifies the minimum number of instruction slots supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinNumberInstructionSlots
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MinNumberTemps Field

Specifies the minimum number of temporary registers supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinNumberTemps
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.MinStaticFlowControlDepth Field

Specifies the minimum depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinStaticFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)








[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PixelShaderCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.PixelShaderCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.PixelShaderCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.PixelShaderCaps instances are equal.
GraphicsDeviceCapabilities.PixelShaderCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PixelShaderCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.PixelShaderCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.PixelShaderCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.op_Equality

Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.PixelShaderCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    PixelShaderCaps left,  
    PixelShaderCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.PixelShaderCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.PixelShaderCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDeviceCapabilities.PixelShaderCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.PixelShaderCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    PixelShaderCaps left,  
    PixelShaderCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.PixelShaderCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.PixelShaderCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.PixelShaderCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)










[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps Properties

Public Properties

	Name	Description
	DynamicFlowControlDepth	Gets a value specifying the dynamic flow control depth.
	NumberInstructionSlots	Gets a value specifying the number of instruction slots supported.
	NumberTemps	Gets a value specifying the number of temporary registers supported.
	StaticFlowControlDepth	Gets a value specifying the depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.
	SupportsArbitrarySwizzle	Gets a value indicating whether arbitrary swizzling is supported.
	SupportsGradientInstructions	Gets a value indicating whether gradient instructions are supported.
	SupportsNoDependentReadLimit	Gets a value indicating whether there is a limit on the number of dependent reads per instruction.
	SupportsNoTextureInstructionLimit	Gets a value indicating whether there is a limit on the number of texture instructions.
	SupportsPredication	Gets a value indicating whether instruction predication is supported.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PixelShaderCaps.DynamicFlowControlDepth Property

Gets a value specifying the dynamic flow control depth.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int DynamicFlowControlDepth { get; }
```

Property Value

The dynamic flow control depth. Must be 0 or 24.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.NumberInstructionSlots Property

Gets a value specifying the number of instruction slots supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int NumberInstructionSlots { get; }
```

Property Value

The number of instruction slots supported.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.NumberTemps Property

Gets a value specifying the number of temporary registers supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int NumberTemps { get; }
```

Property Value

The number of temporary registers supported.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.StaticFlowControlDepth Property

Gets a value specifying the depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StaticFlowControlDepth { get; }
```

Property Value

The static flow control depth.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.SupportsArbitrarySwizzle Property

Gets a value indicating whether arbitrary swizzling is supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsArbitrarySwizzle { get; }
```

Property Value

true if arbitrary swizzling is supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.SupportsGradientInstructions Property

Gets a value indicating whether gradient instructions are supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsGradientInstructions { get; }
```

Property Value

true if gradient instructions are supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.SupportsNoDependentReadLimit Property

Gets a value indicating whether there is a limit on the number of dependent reads per instruction.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNoDependentReadLimit { get; }
```

Property Value

true if there is no dependent read limit; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.SupportsNoTextureInstructionLimit Property

Gets a value indicating whether there is a limit on the number of texture instructions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNoTextureInstructionLimit { get; }
```

Property Value

true if there is no limit on the number of texture instructions; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PixelShaderCaps.SupportsPredication Property

Gets a value indicating whether instruction predication is supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsPredication { get; }
```

Property Value

true if instruction predication is supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PixelShaderCaps Structure](#)

[GraphicsDeviceCapabilities.PixelShaderCaps Members](#)

[PixelShaderCapabilities](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps Structure

Represents driver primitive capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.PrimitiveCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










GraphicsDeviceCapabilities.PrimitiveCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.PrimitiveCaps type.



Public Properties

Name	Description
 HasFogVertexClamped	Gets a value indicating whether the device clamps fog blend factor per vertex.
 IsNullReference	Gets a value indicating whether the device is a null reference device that does not render.
 SupportsBlendOperation	Gets a value indicating whether the device supports blending operations other than BlendFunction.Add .
 SupportsClipTransformedVertices	Gets a value indicating whether the device clips post-transformed vertex primitives.
 SupportsColorWrite	Gets a value indicating whether the device supports per-channel writes for the render-target color buffer through the ColorWriteChannels state.
 SupportsCullClockwiseFace	Gets a value indicating whether the driver supports clockwise triangle culling through the CullMode state.
 SupportsCullCounterClockwiseFace	Gets a value indicating whether the driver supports counterclockwise triangle culling through the CullMode state.
 SupportsCullNone	Gets a value indicating whether the driver does not perform triangle culling.
 SupportsFogAndSpecularAlpha	Gets a value indicating whether the driver supports separate fog and specular alpha.
 SupportsIndependentWriteMasks	Gets a value indicating whether the device supports independent write masks for multiple element textures or multiple render targets.
 SupportsMaskZ	Gets a value indicating whether the device can enable and disable modification of the depth buffer on pixel operations.
 SupportsMultipleRenderTargetIndependentBitDepths	Gets a value indicating whether the device supports different bit depths for multiple render targets.
 SupportsMultipleRenderTargetPostPixelShaderBlending	Gets a value indicating whether the device supports post-pixel shader operations for multiple render targets.
 SupportsSeparateAlphaBlend	Gets a value indicating whether the device supports separate blend settings for the alpha channel.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
  Op_Equality	Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are equal.
  Op_Inequality	Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PrimitiveCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PrimitiveCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.PrimitiveCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.PrimitiveCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.PrimitiveCaps instances are equal.
GraphicsDeviceCapabilities.PrimitiveCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps](#) Structure

[GraphicsDeviceCapabilities.PrimitiveCaps](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

GraphicsDeviceCapabilities.PrimitiveCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.PrimitiveCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.PrimitiveCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.PrimitiveCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    PrimitiveCaps left,  
    PrimitiveCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.PrimitiveCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.PrimitiveCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.PrimitiveCaps.op_Inequality

Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.PrimitiveCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    PrimitiveCaps left,  
    PrimitiveCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.PrimitiveCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.PrimitiveCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDeviceCapabilities.PrimitiveCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)















[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps Properties

Public Properties

Name	Description
 HasFogVertexClamped	Gets a value indicating whether the device clamps fog blend factor per vertex.
 IsNullReference	Gets a value indicating whether the device is a null reference device that does not render.
 SupportsBlendOperation	Gets a value indicating whether the device supports blending operations other than BlendFunction.Add .
 SupportsClipTransformedVertices	Gets a value indicating whether the device clips post-transformed vertex primitives.
 SupportsColorWrite	Gets a value indicating whether the device supports per-channel writes for the render-target color buffer through the ColorWriteChannels state.
 SupportsCullClockwiseFace	Gets a value indicating whether the driver supports clockwise triangle culling through the CullMode state.
 SupportsCullCounterClockwiseFace	Gets a value indicating whether the driver supports counterclockwise triangle culling through the CullMode state.
 SupportsCullNone	Gets a value indicating whether the driver does not perform triangle culling.
 SupportsFogAndSpecularAlpha	Gets a value indicating whether the driver supports separate fog and specular alpha.
 SupportsIndependentWriteMasks	Gets a value indicating whether the device supports independent write masks for multiple element textures or multiple render targets.
 SupportsMaskZ	Gets a value indicating whether the device can enable and disable modification of the depth buffer on pixel operations.
 SupportsMultipleRenderTargetsIndependentBitDepths	Gets a value indicating whether the device supports different bit depths for multiple render targets.
 SupportsMultipleRenderTargetsPostPixelShaderBlending	Gets a value indicating whether the device supports post-pixel shader operations for multiple render targets.
 SupportsSeparateAlphaBlend	Gets a value indicating whether the device supports separate blend settings for the alpha channel.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.PrimitiveCaps.HasFogVertexClamped Property

Gets a value indicating whether the device clamps fog blend factor per vertex.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasFogVertexClamped { get; }
```

Property Value

true if the device clamps fog blend factor per vertex; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.IsNullReference Property

Gets a value indicating whether the device is a null reference device that does not render.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsNullReference { get; }
```

Property Value

true if the device is a null reference device; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsBlendOperation Property

Gets a value indicating whether the device supports blending operations other than [BlendFunction.Add](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsBlendOperation { get; }
```

Property Value

true if the driver supports other blend operations; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsClipTransformedVertices Property

Gets a value indicating whether the device clips post-transformed vertex primitives.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsClipTransformedVertices { get; }
```

Property Value

true if the device clips post-transformed vertex primitives; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsColorWrite Property

Gets a value indicating whether the device supports per-channel writes for the render-target color buffer through the [ColorWriteChannels](#) state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsColorWrite { get; }
```

Property Value

true if the device supports per-channel writes; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsCullClockwiseFace Property

Gets a value indicating whether the driver supports clockwise triangle culling through the [CullMode](#) state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsCullClockwiseFace { get; }
```

Property Value

true if the driver supports clockwise triangle culling; **false** otherwise.

Remarks

This applies only to triangle primitives.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsCullCounterClockwiseFace Property

Gets a value indicating whether the driver supports counterclockwise triangle culling through the [CullMode](#) state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsCullCounterClockwiseFace { get; }
```

Property Value

true if the driver supports counterclockwise triangle culling; **false** otherwise.

Remarks

This applies only to triangle primitives.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsCullNone Property

Gets a value indicating whether the driver does not perform triangle culling.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsCullNone { get; }
```

Property Value

true if the driver does not perform triangle culling; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsFogAndSpecularAlpha Property

Gets a value indicating whether the driver supports separate fog and specular alpha.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFogAndSpecularAlpha { get; }
```

Property Value

true if the driver supports separate fog and specular alpha; **false** otherwise.

Remarks

Many devices use the specular alpha channel to store the fog factor.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsIndependentWriteMasks Property

Gets a value indicating whether the device supports independent write masks for multiple element textures or multiple render targets.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsIndependentWriteMasks { get; }
```

Property Value

true if the device supports independent write masks; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsMaskZ Property

Gets a value indicating whether the device can enable and disable modification of the depth buffer on pixel operations.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMaskZ { get; }
```

Property Value

true if the device can enable and disable modification of the depth buffer; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsMultipleRenderTargetsIndependentBitDepths Property

Gets a value indicating whether the device supports different bit depths for multiple render targets.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMultipleRenderTargetsIndependentBitDepths { get; }
```

Property Value

true if the device supports different bit depths for multiple render targets; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsMultipleRenderTargetsPostPixelShaderBlending Property

Gets a value indicating whether the device supports post-pixel shader operations for multiple render targets.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMultipleRenderTargetsPostPixelShaderBlending { get; }
```

Property Value

true if the device supports post-pixel shader operations for multiple render targets; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.PrimitiveCaps.SupportsSeparateAlphaBlend Property

Gets a value indicating whether the device supports separate blend settings for the alpha channel.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSeparateAlphaBlend { get; }
```

Property Value

true if the device supports separate blend settings for the alpha channel; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.PrimitiveCaps Structure](#)

[GraphicsDeviceCapabilities.PrimitiveCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps Structure

Represents raster-drawing capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.RasterCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Members](#)















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.RasterCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.RasterCaps type.



Public Properties

Name	Description
 SupportsAnisotropy	Gets a value indicating whether the device supports anisotropic filtering.
 SupportsColorPerspective	Gets a value indicating whether the device iterates colors perspective correctly.
 SupportsDepthBias	Gets a value indicating whether the device supports legacy depth bias.
 SupportsDepthBufferLessHsr	Gets a value indicating whether the device can perform hidden-surface removal (HSR) without requiring the application to sort polygons and without requiring the allocation of a depth buffer.
 SupportsDepthBufferTest	Gets a value indicating whether the device can perform depth-test operations.
 SupportsDepthFog	Gets a value indicating whether the device supports depth-based fog.
 SupportsFogRange	Gets a value indicating whether the device supports range-based fog.
 SupportsFogTable	Gets a value indicating whether the device calculates the fog value by referring to a lookup table containing fog values that are indexed to the depth of a given pixel.
 SupportsFogVertex	Gets a value indicating whether the device calculates the fog value during the lighting operation and interpolates the fog value during rasterization.
 SupportsMipMapLevelOfDetailBias	Gets a value indicating whether the device supports level-of-detail bias adjustments.
 SupportsMultisampleToggle	Gets a value indicating whether the device supports toggling multisampling on and off.
 SupportsScissorTest	Gets a value indicating whether the device supports scissor test.
 SupportsSlopeScaleDepthBias	Gets a value indicating whether the device performs true slope-scale based depth bias.
 SupportsWFog	Gets a value indicating whether the device supports w-based fog.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.RasterCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.RasterCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.RasterCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.RasterCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.RasterCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.RasterCaps instances are equal.
GraphicsDeviceCapabilities.RasterCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.RasterCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.RasterCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.RasterCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.RasterCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    RasterCaps left,  
    RasterCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.RasterCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.RasterCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.RasterCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.RasterCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    RasterCaps left,  
    RasterCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.RasterCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.RasterCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.RasterCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)















[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps Properties

Public Properties

	Name	Description
	SupportsAnisotropy	Gets a value indicating whether the device supports anisotropic filtering.
	SupportsColorPerspective	Gets a value indicating whether the device iterates colors perspective correctly.
	SupportsDepthBias	Gets a value indicating whether the device supports legacy depth bias.
	SupportsDepthBufferLessHsr	Gets a value indicating whether the device can perform hidden-surface removal (HSR) without requiring the application to sort polygons and without requiring the allocation of a depth buffer.
	SupportsDepthBufferTest	Gets a value indicating whether the device can perform depth-test operations.
	SupportsDepthFog	Gets a value indicating whether the device supports depth-based fog.
	SupportsFogRange	Gets a value indicating whether the device supports range-based fog.
	SupportsFogTable	Gets a value indicating whether the device calculates the fog value by referring to a lookup table containing fog values that are indexed to the depth of a given pixel.
	SupportsFogVertex	Gets a value indicating whether the device calculates the fog value during the lighting operation and interpolates the fog value during rasterization.
	SupportsMipMapLevelOfDetailBias	Gets a value indicating whether the device supports level-of-detail bias adjustments.
	SupportsMultisampleToggle	Gets a value indicating whether the device supports toggling multisampling on and off.
	SupportsScissorTest	Gets a value indicating whether the device supports scissor test.
	SupportsSlopeScaleDepthBias	Gets a value indicating whether the device performs true slope-scale based depth bias.
	SupportsWFog	Gets a value indicating whether the device supports w-based fog.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.RasterCaps.SupportsAnisotropy Property

Gets a value indicating whether the device supports anisotropic filtering.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAnisotropy { get; }
```

Property Value

true if the device supports anisotropic filtering; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsColorPerspective Property

Gets a value indicating whether the device iterates colors perspective correctly.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsColorPerspective { get; }
```

Property Value

true if the device iterates colors perspective correctly; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsDepthBias Property

Gets a value indicating whether the device supports legacy depth bias.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDepthBias { get; }
```

Property Value

true if the device supports legacy depth bias; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsDepthBufferLessHsr Property

Gets a value indicating whether the device can perform hidden-surface removal (HSR) without requiring the application to sort polygons and without requiring the allocation of a depth buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDepthBufferLessHsr { get; }
```

Property Value

true if the device can perform hidden-surface removal (HSR) without requiring the application to sort polygons and without requiring the allocation of a depth buffer; **false** otherwise.

Remarks

This leaves more video memory for textures. The method used to perform HSR is hardware-dependent and is transparent to the application.

Z-bufferless HSR is performed if no depth-buffer surface is associated with the rendering-target surface and the depth-buffer comparison test is enabled (that is, when the state value associated with the [DepthBufferEnable](#) is true).

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsDepthBufferTest Property

Gets a value indicating whether the device can perform depth-test operations.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDepthBufferTest { get; }
```

Property Value

true if the device can perform depth-test operations; **false** otherwise.

Remarks

This effectively renders a primitive and indicates whether any depth pixels have been rendered.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsDepthFog Property

Gets a value indicating whether the device supports depth-based fog.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDepthFog { get; }
```

Property Value

true if the device supports depth-based fog; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsFogRange Property

Gets a value indicating whether the device supports range-based fog.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFogRange { get; }
```

Property Value

true if the device supports range-based fog; **false** otherwise.

Remarks

In range-based fog, the distance of an object from the viewer is used to compute fog effects, not the depth of the object (that is, the z-coordinate) in the scene.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsFogTable Property

Gets a value indicating whether the device calculates the fog value by referring to a lookup table containing fog values that are indexed to the depth of a given pixel.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFogTable { get; }
```

Property Value

true if the device calculates the fog value by referring to a lookup table; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsFogVertex Property

Gets a value indicating whether the device calculates the fog value during the lighting operation and interpolates the fog value during rasterization.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFogVertex { get; }
```

Property Value

true if the device calculates the fog value during the lighting operation and interpolates the fog value during rasterization; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsMipMapLevelOfDetailBias Property

Gets a value indicating whether the device supports level-of-detail bias adjustments.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMipMapLevelOfDetailBias { get; }
```

Property Value

true if the device supports level-of-detail bias adjustments; **false** otherwise.

Remarks

These bias adjustments enable an application to make a mipmap appear crisper or less sharp than it normally would.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsMultisampleToggle Property

Gets a value indicating whether the device supports toggling multisampling on and off.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMultisampleToggle { get; }
```

Property Value

true if the device supports toggling multisampling on and off; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsScissorTest Property

Gets a value indicating whether the device supports scissor test.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsScissorTest { get; }
```

Property Value

true if the device supports scissor test; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsSlopeScaleDepthBias Property

Gets a value indicating whether the device performs true slope-scale based depth bias.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSlopeScaleDepthBias { get; }
```

Property Value

true if the device performs true slope-scale based depth bias; **false** otherwise.

Remarks

This is in contrast to the legacy-style depth bias.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.RasterCaps.SupportsWFog Property

Gets a value indicating whether the device supports w-based fog.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsWFog { get; }
```

Property Value

true if the device supports w-based fog; **false** otherwise.

Remarks

W-based fog is used when a perspective projection matrix is specified, but affine projections still use z-based fog. The system considers a projection matrix that contains a nonzero value in the [3][4] element to be a perspective projection matrix.

See Also

Reference

[GraphicsDeviceCapabilities.RasterCaps Structure](#)

[GraphicsDeviceCapabilities.RasterCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps Structure

Represents shading operations capabilities

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.ShadingCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Members](#)





[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.ShadingCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.ShadingCaps type.



Public Properties

Name	Description
 SupportsAlphaGouraudBlend	Gets a value indicating whether the device can support an alpha component for Gouraud-blended transparency.
 SupportsColorGouraudRgb	Gets a value indicating whether the device can support colored Gouraud shading.
 SupportsFogGouraud	Gets a value indicating whether the device can support fog in the Gouraud shading mode.
 SupportsSpecularGouraudRgb	Gets a value indicating whether the device supports Gouraud shading of specular highlights.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 S op_Equality	Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are equal.
 S op_Inequality	Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.ShadingCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.ShadingCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.ShadingCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.ShadingCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.ShadingCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.ShadingCaps instances are equal.
GraphicsDeviceCapabilities.ShadingCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps](#) Structure

[GraphicsDeviceCapabilities.ShadingCaps](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

GraphicsDeviceCapabilities.ShadingCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.ShadingCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.ShadingCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.ShadingCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    ShadingCaps left,  
    ShadingCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.ShadingCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.ShadingCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps](#) Structure

[GraphicsDeviceCapabilities.ShadingCaps](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.ShadingCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.ShadingCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    ShadingCaps left,  
    ShadingCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.ShadingCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.ShadingCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps](#) Structure

[GraphicsDeviceCapabilities.ShadingCaps](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.ShadingCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)





[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps Properties

Public Properties

	Name	Description
	SupportsAlphaGouraudBlend	Gets a value indicating whether the device can support an alpha component for Gouraud-blended transparency.
	SupportsColorGouraudRgb	Gets a value indicating whether the device can support colored Gouraud shading.
	SupportsFogGouraud	Gets a value indicating whether the device can support fog in the Gouraud shading mode.
	SupportsSpecularGouraudRgb	Gets a value indicating whether the device supports Gouraud shading of specular highlights.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.ShadingCaps.SupportsAlphaGouraudBlend Property

Gets a value indicating whether the device can support an alpha component for Gouraud-blended transparency.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAlphaGouraudBlend { get; }
```

Property Value

true if can support an alpha component for Gouraud-blended transparency; **false** otherwise.

Remarks

In this mode, the alpha color component of a primitive is provided at vertices and interpolated across a face along with the other color components.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps.SupportsColorGouraudRgb Property

Gets a value indicating whether the device can support colored Gouraud shading.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsColorGouraudRgb { get; }
```

Property Value

true if the device can support colored Gouraud shading; **false** otherwise.

Remarks

In this mode, the per-vertex color components (red, green, and blue) are interpolated across a triangle face.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps.SupportsFogGouraud Property

Gets a value indicating whether the device can support fog in the Gouraud shading mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsFogGouraud { get; }
```

Property Value

true if the device can support fog in the Gouraud shading mode; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.ShadingCaps.SupportsSpecularGouraudRgb Property

Gets a value indicating whether the device supports Gouraud shading of specular highlights.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsSpecularGouraudRgb { get; }
```

Property Value

true if the device supports Gouraud shading of specular highlights; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.ShadingCaps Structure](#)

[GraphicsDeviceCapabilities.ShadingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps Structure

Represents driver stencil capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.StencilCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Members](#)










[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.StencilCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.StencilCaps type.



Public Properties

Name	Description
 SupportsDecrement	Gets a value indicating whether the device supports decrementing the stencil-buffer entry, wrapping to the maximum value if the new value is less than zero.
 SupportsDecrementSaturation	Gets a value indicating whether the device supports decrementing the stencil-buffer entry, clamping to zero.
 SupportsIncrement	Gets a value indicating whether the device supports incrementing the stencil-buffer entry, wrapping to zero if the new value exceeds the maximum value.
 SupportsIncrementSaturation	Gets a value indicating whether the device supports incrementing the stencil-buffer entry, clamping to the maximum value.
 SupportsInvert	Gets a value indicating whether the device supports inverting the bits in the stencil-buffer entry.
 SupportsKeep	Gets a value indicating whether the device does not update the entry in the stencil buffer.
 SupportsReplace	Gets a value indicating whether the device supports replacing the stencil-buffer entry with a reference value.
 SupportsTwoSided	Gets a value indicating whether the device supports two-sided stencil.
 SupportsZero	Gets a value indicating whether the device supports setting the stencil-buffer entry to 0.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are equal.
 Op_Inequality	Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also








Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.StencilCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are equal.
	Op_Inequality	Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.StencilCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.StencilCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.StencilCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.StencilCaps instances are equal.
GraphicsDeviceCapabilities.StencilCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.StencilCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.StencilCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.StencilCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.StencilCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    StencilCaps left,  
    StencilCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.StencilCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.StencilCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.StencilCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.StencilCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    StencilCaps left,  
    StencilCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.StencilCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.StencilCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.StencilCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)










[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps Properties

Public Properties

	Name	Description
	SupportsDecrement	Gets a value indicating whether the device supports decrementing the stencil-buffer entry, wrapping to the maximum value if the new value is less than zero.
	SupportsDecrementSaturation	Gets a value indicating whether the device supports decrementing the stencil-buffer entry, clamping to zero.
	SupportsIncrement	Gets a value indicating whether the device supports incrementing the stencil-buffer entry, wrapping to zero if the new value exceeds the maximum value.
	SupportsIncrementSaturation	Gets a value indicating whether the device supports incrementing the stencil-buffer entry, clamping to the maximum value.
	SupportsInvert	Gets a value indicating whether the device supports inverting the bits in the stencil-buffer entry.
	SupportsKeep	Gets a value indicating whether the device does not update the entry in the stencil buffer.
	SupportsReplace	Gets a value indicating whether the device supports replacing the stencil-buffer entry with a reference value.
	SupportsTwoSided	Gets a value indicating whether the device supports two-sided stencil.
	SupportsZero	Gets a value indicating whether the device supports setting the stencil-buffer entry to 0.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.StencilCaps.SupportsDecrement Property

Gets a value indicating whether the device supports decrementing the stencil-buffer entry, wrapping to the maximum value if the new value is less than zero.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDecrement { get; }
```

Property Value

true if the device supports decrementing the stencil-buffer entry; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsDecrementSaturation Property

Gets a value indicating whether the device supports decrementing the stencil-buffer entry, clamping to zero.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDecrementSaturation { get; }
```

Property Value

true if the device supports the stencil-buffer entry, clamping to zero; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsIncrement Property

Gets a value indicating whether the device supports incrementing the stencil-buffer entry, wrapping to zero if the new value exceeds the maximum value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsIncrement { get; }
```

Property Value

true if the device supports incrementing the stencil-buffer entry, wrapping to zero if the new value exceeds the maximum value; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsIncrementSaturation Property

Gets a value indicating whether the device supports incrementing the stencil-buffer entry, clamping to the maximum value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsIncrementSaturation { get; }
```

Property Value

true if the device supports incrementing the stencil-buffer entry, clamping to the maximum value; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsInvert Property

Gets a value indicating whether the device supports inverting the bits in the stencil-buffer entry.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsInvert { get; }
```

Property Value

true if the device supports inverting the bits in the stencil-buffer entry; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsKeep Property

Gets a value indicating whether the device does not update the entry in the stencil buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsKeep { get; }
```

Property Value

true if the device does not update the entry in the stencil buffer; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsReplace Property

Gets a value indicating whether the device supports replacing the stencil-buffer entry with a reference value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsReplace { get; }
```

Property Value

true if the device supports replacing the stencil-buffer entry with a reference value; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsTwoSided Property

Gets a value indicating whether the device supports two-sided stencil.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTwoSided { get; }
```

Property Value

true if the device supports two-sided stencil; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.StencilCaps.SupportsZero Property

Gets a value indicating whether the device supports setting the stencil-buffer entry to 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsZero { get; }
```

Property Value

true if the device supports setting the stencil-buffer entry to 0; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.StencilCaps Structure](#)

[GraphicsDeviceCapabilities.StencilCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps Structure

Represents miscellaneous texture-mapping capabilities

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.TextureCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Members](#)
















[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.TextureCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.TextureCaps type.



Public Properties

Name	Description
 RequiresCubeMapPower2	Gets a value indicating whether the device requires that cube texture maps have dimensions specified as powers of two.
 RequiresPower2	Gets a value indicating whether the device only supports textures that are powers of two.
 RequiresSquareOnly	Gets a value indicating whether the device requires all textures to be square.
 RequiresVolumeMapPower2	Gets a value indicating whether the device requires that volume texture maps have dimensions specified as powers of two.
 SupportsAlpha	Gets a value indicating whether the device supports alpha in texture pixels.
 SupportsCubeMap	Gets a value indicating whether the device supports cube textures
 SupportsMipCubeMap	Gets a value indicating whether the device supports mipmapped cube textures.
 SupportsMipMap	Gets a value indicating whether the device supports mipmapped textures.
 SupportsMipVolumeMap	Gets a value indicating whether the device supports mipmapped volume textures.
 SupportsNonPower2Conditional	Gets a value indicating whether the device supports the use of 2D textures with dimensions that are not powers of two, under certain conditions.
 SupportsNoProjectedBumpEnvironment	Gets a value indicating whether the device does not support a projected bump-environment look up operation in programmable and fixed-function shaders.
 SupportsPerspective	Gets a value indicating whether the device supports perspective correction texturing
 SupportsProjected	Gets a value indicating whether the device supports per pixel projective divide.
 SupportsTextureRepeatNotScaledBySize	Gets a value indicating whether the device does not scale texture indices by the texture size prior to interpolation.
 SupportsVolumeMap	Gets a value indicating whether the device supports volume textures.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are equal.
 GetHashCode	Gets the hash code for this object.
 GetType	(Inherited from Object .)
 S op_Equality	Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are equal.
 S op_Inequality	Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)








See Also

Reference



[GraphicsDeviceCapabilities.TextureCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.TextureCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.TextureCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.TextureCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.TextureCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.TextureCaps instances are equal.
GraphicsDeviceCapabilities.TextureCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.TextureCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.TextureCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.TextureCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.TextureCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    TextureCaps left,  
    TextureCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.TextureCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.TextureCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.TextureCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.TextureCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    TextureCaps left,  
    TextureCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.TextureCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.TextureCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.TextureCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)
















[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps Properties

Public Properties

	Name	Description
	RequiresCubeMapPower2	Gets a value indicating whether the device requires that cube texture maps have dimensions specified as powers of two.
	RequiresPower2	Gets a value indicating whether the device only supports textures that are powers of two.
	RequiresSquareOnly	Gets a value indicating whether the device requires all textures to be square.
	RequiresVolumeMapPower2	Gets a value indicating whether the device requires that volume texture maps have dimensions specified as powers of two.
	SupportsAlpha	Gets a value indicating whether the device supports alpha in texture pixels.
	SupportsCubeMap	Gets a value indicating whether the device supports cube textures
	SupportsMipCubeMap	Gets a value indicating whether the device supports mipmapped cube textures.
	SupportsMipMap	Gets a value indicating whether the device supports mipmapped textures.
	SupportsMipVolumeMap	Gets a value indicating whether the device supports mipmapped volume textures.
	SupportsNonPower2Conditional	Gets a value indicating whether the device supports the use of 2D textures with dimensions that are not powers of two, under certain conditions.
	SupportsNoProjectedBumpEnvironment	Gets a value indicating whether the device does not support a projected bump-environment look up operation in programmable and fixed-function shaders.
	SupportsPerspective	Gets a value indicating whether the device supports perspective correction texturing
	SupportsProjected	Gets a value indicating whether the device supports per pixel projective divide.
	SupportsTextureRepeatNotScaledBySize	Gets a value indicating whether the device does not scale texture indices by the texture size prior to interpolation.
	SupportsVolumeMap	Gets a value indicating whether the device supports volume textures.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.TextureCaps.RequiresCubeMapPower2 Property

Gets a value indicating whether the device requires that cube texture maps have dimensions specified as powers of two.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool RequiresCubeMapPower2 { get; }
```

Property Value

true if the device requires that cube texture maps have dimensions specified as powers of two; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.RequiresPower2 Property

Gets a value indicating whether the device only supports textures that are powers of two.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool RequiresPower2 { get; }
```

Property Value

true if the device only supports textures that are powers of two; **false** otherwise.

Remarks

The effect of [RequiresPower2](#) also depends on the value of [SupportsNonPower2Conditional](#):

RequiresPower2 value	SupportsNonPower2Conditional value	Result
false	false	Unconditional support for 2D textures with dimensions that are not powers of 2.
true	false	All textures must have widths and heights specified as powers of 2. This requirement does not apply to either cube textures or volume textures.
true	true	The device conditionally supports the use of 2D textures with dimensions that are not powers of 2. Mipmaps and wrapping are not supported. For more information, see SupportsNonPower2Conditional .

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[SupportsNonPower2Conditional](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.RequiresSquareOnly Property

Gets a value indicating whether the device requires all textures to be square.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool RequiresSquareOnly { get; }
```

Property Value

true if the device requires all textures to be square; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.RequiresVolumeMapPower2 Property

Gets a value indicating whether the device requires that volume texture maps have dimensions specified as powers of two.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool RequiresVolumeMapPower2 { get; }
```

Property Value

true if the device requires that volume texture maps have dimensions specified as powers of two; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsAlpha Property

Gets a value indicating whether the device supports alpha in texture pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsAlpha { get; }
```

Property Value

true if the device supports alpha in texture pixels; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsCubeMap Property

Gets a value indicating whether the device supports cube textures

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsCubeMap { get; }
```

Property Value

true if the device supports cube textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsMipCubeMap Property

Gets a value indicating whether the device supports mipmapped cube textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMipCubeMap { get; }
```

Property Value

true if the device supports mipmapped cube textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsMipMap Property

Gets a value indicating whether the device supports mipmapped textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMipMap { get; }
```

Property Value

true if the device supports mipmapped textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsMipVolumeMap Property

Gets a value indicating whether the device supports mipmapped volume textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsMipVolumeMap { get; }
```

Property Value

true if the device supports mipmapped volume textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsNonPower2Conditional Property

Gets a value indicating whether the device supports the use of 2D textures with dimensions that are not powers of two, under certain conditions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNonPower2Conditional { get; }
```

Property Value

true if the device supports the use of 2D textures with dimensions that are not powers of two, under certain conditions; **false** otherwise.

Remarks

The device can use a texture with dimensions that are not a power of two, as long as these conditions are met:

- The texture-addressing mode for the texture stage is set to [TextureAddressMode.Clamp](#).
- Texture wrapping for the texture stage is disabled ([Wrap1](#), and [Wrap15](#) is set to [TextureWrapCoordinates.Zero](#)).
- Mipmapping is not in use (use magnification filter only).
- Texture formats must not be [SurfaceFormat.Dxt1](#) through [SurfaceFormat.Dxt5](#).

If this flag is not set, and [RequiresPower2](#) also is not set, then unconditional support is provided for 2D textures with dimensions that are not powers of two.

A texture that is not a power of two cannot be set at a stage that will be read based on a shader computation (such as the **bem** - ps and **texm3x3** - ps instructions in pixel shaders versions 1_0 to 1_3). For example, these textures can be used to store bumps that will be fed into texture reads, but not the environment maps that are used in **texbem** - ps, **texbem1** - ps, and **texm3x3spec** - ps. This means that a texture with dimensions that are not powers of two cannot be addressed or sampled using texture coordinates computed within the shader. This type of operation is known as a dependent read. It cannot be performed on these types of textures.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[RequiresPower2](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsNoProjectedBumpEnvironment Property

Gets a value indicating whether the device does not support a projected bump-environment lookup operation in programmable and fixed-function shaders.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNoProjectedBumpEnvironment { get; }
```

Property Value

true if the device does not support a projected bump-environment lookup operation in programmable and fixed-function shaders; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsPerspectiveProperty

Gets a value indicating whether the device supports perspective correction texturing

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsPerspective { get; }
```

Property Value

true if the device supports perspective correction texturing; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsProjectedProperty

Gets a value indicating whether the device supports per pixel projective divide.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsProjected { get; }
```

Property Value

true if the device supports supports per pixel projective divide; **false** otherwise.

Remarks

When applied, the device divides transformed texture coordinates by the last texture coordinate. If this capability is present, the projective divide occurs per pixel. If this capability is not present, but the projective divide needs to occur anyway, it is performed on a per-vertex basis by the Direct3D runtime.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsTextureRepeatNotScaledBySize Property

Gets a value indicating whether the device does not scale texture indices by the texture size prior to interpolation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureRepeatNotScaledBySize { get; }
```

Property Value

true if the device does not scale texture indices by the texture size prior to interpolation; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.TextureCaps.SupportsVolumeMap Property

Gets a value indicating whether the device supports volume textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsVolumeMap { get; }
```

Property Value

true if the device supports volume textures; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.TextureCaps Structure](#)

[GraphicsDeviceCapabilities.TextureCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCaps Structure

Represents flexible vertex format capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.VertexFormatCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)




[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










GraphicsDeviceCapabilities.VertexFormatCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.VertexFormatCaps type.



Public Properties

	Name	Description
	NumberSimultaneousTextureCoordinates	Gets the total number of texture coordinate sets that the device can simultaneously use for multiple texture blending.
	SupportsDoNotStripElements	Gets a value indicating whether vertex elements should not be stripped.
	SupportsPointSize	Gets a value indicating whether point size comes from point size data in the vertex declaration.

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are equal.
 	op_Inequality	Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)










See Also

Reference



[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexFormatCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are equal.
 	op_Inequality	Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexFormatCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.VertexFormatCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.VertexFormatCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.VertexFormatCaps instances are equal.
GraphicsDeviceCapabilities.VertexFormatCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexFormatCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.VertexFormatCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.VertexFormatCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.VertexFormatCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexFormatCaps left,  
    VertexFormatCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.VertexFormatCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.VertexFormatCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDeviceCapabilities.VertexFormatCaps.op_Inequality

Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.VertexFormatCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexFormatCaps left,  
    VertexFormatCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.VertexFormatCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.VertexFormatCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.VertexFormatCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)




[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCaps Properties

Public Properties

	Name	Description
	NumberSimultaneousTextureCoordinates	Gets the total number of texture coordinate sets that the device can simultaneously use for multiple texture blending.
	SupportsDoNotStripElements	Gets a value indicating whether vertex elements should not be stripped.
	SupportsPointSize	Gets a value indicating whether point size comes from point size data in the vertex declaration.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexFormatCaps.NumberSimultaneousTextureCoordinates Property

Gets the total number of texture coordinate sets that the device can simultaneously use for multiple texture blending.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public short NumberSimultaneousTextureCoordinates { get; }
```

Property Value

The total number of texture coordinate sets that the device can simultaneously use for multiple texture blending. (You can use up to eight texture coordinate sets for any vertex, but the device can blend using only the specified number of texture coordinate sets.)

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCaps.SupportsDoNotStripElements Property

Gets a value indicating whether vertex elements should not be stripped.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsDoNotStripElements { get; }
```

Property Value

true if vertex elements should not be stripped; otherwise false.

Remarks

If the vertex format contains elements that are not used with the current render states, there is no need to regenerate the vertices. If this property is false, stripping extraneous elements from the vertex format provides better performance.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexFormatCaps.SupportsPointSize Property

Gets a value indicating whether point size comes from point size data in the vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsPointSize { get; }
```

Property Value

true if point size comes from point size data in the vertex declaration; otherwise false.

Remarks

If this property is false, point size is determined by [PointSize](#). If an application provides point size in both the render state and the vertex declaration, the vertex data overrides the render-state data.

See Also

Reference

[GraphicsDeviceCapabilities.VertexFormatCaps Structure](#)

[GraphicsDeviceCapabilities.VertexFormatCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps Structure

Represents vertex processing capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.VertexProcessingCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)





[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GraphicsDeviceCapabilities.VertexProcessingCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.VertexProcessingCaps type.



Public Properties

	Name	Description
	SupportsLocalViewer	Gets a value indicating whether the device can do local viewer.
	SupportsNoTextureGenerationNonLocalViewer	Gets a value indicating whether the device supports texture generation in non-local viewer mode.
	SupportsTextureGeneration	Gets a value indicating whether the device can do texture generation.
	SupportsTextureGenerationSphereMap	Gets a value indicating whether the device supports use of the specified texture coordinates for sphere mapping.

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are equal.
	op_Inequality	Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)










See Also

Reference



[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexProcessingCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are equal.
	GetHashCode	Gets the hash code for this object.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are equal.
 	op_Inequality	Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexProcessingCaps.Equals

Method

Determines whether two [GraphicsDeviceCapabilities.VertexProcessingCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.VertexProcessingCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.VertexProcessingCaps instances are equal.
GraphicsDeviceCapabilities.VertexProcessingCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexProcessingCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.VertexProcessingCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.VertexProcessingCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps.GetHashCode Method

Gets the hash code for this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

The hash code for this object.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.VertexProcessingCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexProcessingCaps left,  
    VertexProcessingCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.VertexProcessingCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.VertexProcessingCaps](#) on the right side of the equal sign.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GraphicsDeviceCapabilities.VertexProcessingCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.VertexProcessingCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexProcessingCaps left,  
    VertexProcessingCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.VertexProcessingCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.VertexProcessingCaps](#) on the right side of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.VertexProcessingCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)





[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps Properties

Public Properties

	Name	Description
	SupportsLocalViewer	Gets a value indicating whether the device can do local viewer.
	SupportsNoTextureGenerationNonLocalViewer	Gets a value indicating whether the device supports texture generation in non-local viewer mode.
	SupportsTextureGeneration	Gets a value indicating whether the device can do texture generation.
	SupportsTextureGenerationSphereMap	Gets a value indicating whether the device supports use of the specified texture coordinates for sphere mapping.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexProcessingCaps.SupportsLocalViewer Property

Gets a value indicating whether the device can do local viewer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsLocalViewer { get; }
```

Property Value

true if the device can do local viewer; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps.SupportsNoTextureGenerationNonLocalViewer Property

Gets a value indicating whether the device supports texture generation in non-local viewer mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsNoTextureGenerationNonLocalViewer { get; }
```

Property Value

true if the device does not support texture generation in non-local viewer mode; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps.SupportsTextureGeneration Property

Gets a value indicating whether the device can do texture generation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureGeneration { get; }
```

Property Value

true if the device can do texture generation; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexProcessingCaps.SupportsTextureGenerationSphereMap Property

Gets a value indicating whether the device supports use of the specified texture coordinates for sphere mapping.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsTextureGenerationSphereMap { get; }
```

Property Value

true if the device supports use of the specified texture coordinates for sphere mapping; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexProcessingCaps Structure](#)

[GraphicsDeviceCapabilities.VertexProcessingCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps Structure

Represents vertex shader version 2_0 extended capabilities.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GraphicsDeviceCapabilities.VertexShaderCaps
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)







[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





GraphicsDeviceCapabilities.VertexShaderCaps Members

The following tables list the members exposed by the GraphicsDeviceCapabilities.VertexShaderCaps type.







Public Fields

Name	Description
 MaxDynamicFlowControlDepth	Specifies the maximum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if_pred - vs).
 MaxNumberTemps	Specifies the maximum number of temporary registers supported.
 MaxStaticFlowControlDepth	Specifies the maximum depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.
 MinDynamicFlowControlDepth	Specifies the minimum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if_pred - vs).
 MinNumberTemps	Specifies the minimum number of temporary registers supported.
 MinStaticFlowControlDepth	Specifies the minimum depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.



Public Properties

Name	Description
 DynamicFlowControlDepth	Gets a value specifying the depth of the dynamic flow control instruction nesting.
 NumberTemps	Gets a value specifying the number of temporary registers supported.
 StaticFlowControlDepth	Gets a value specifying the depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.
 SupportsPredication	Gets a value indicating whether instruction predication is supported.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are equal.
 GetType	(Inherited from Object .)
 op_Equality	Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are equal.
 op_Inequality	Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are unequal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also







Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexShaderCaps Fields

Public Fields

	Name	Description
	MaxDynamicFlowControlDepth	Specifies the maximum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if pred - vs).
	MaxNumberTemps	Specifies the maximum number of temporary registers supported.
	MaxStaticFlowControlDepth	Specifies the maximum depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.
	MinDynamicFlowControlDepth	Specifies the minimum level of nesting of dynamic flow control instructions (break - vs, break_comp - vs, breakp - vs, if_comp - vs, if_comp - vs, if pred - vs).
	MinNumberTemps	Specifies the minimum number of temporary registers supported.
	MinStaticFlowControlDepth	Specifies the minimum depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexShaderCaps.MaxDynamicFlowControlDepth Field

Specifies the maximum level of nesting of dynamic flow control instructions (**break - vs**, **break_comp - vs**, **breakp - vs**, **if_comp - vs**, **if_comp - vs**, **if_pred - vs**).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxDynamicFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.MaxNumberTemps Field

Specifies the maximum number of temporary registers supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxNumberTemps
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.MaxStaticFlowControlDepth Field

Specifies the maximum depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxStaticFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.MinDynamicFlowControlDepth Field

Specifies the minimum level of nesting of dynamic flow control instructions (**break - vs**, **break_comp - vs**, **breakp - vs**, **if_comp - vs**, **if_comp - vs**, **if_pred - vs**).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinDynamicFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.MinNumberTemps Field

Specifies the minimum number of temporary registers supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinNumberTemps
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.MinStaticFlowControlDepth Field

Specifies the minimum depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MinStaticFlowControlDepth
```

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)









[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are equal.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are equal.
 	op_Inequality	Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are unequal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexShaderCaps.Equals Method

Determines whether two [GraphicsDeviceCapabilities.VertexShaderCaps](#) instances are equal.

Overload List

Name	Description
GraphicsDeviceCapabilities.VertexShaderCaps.Equals (Object)	Determines whether two GraphicsDeviceCapabilities.VertexShaderCaps instances are equal.
GraphicsDeviceCapabilities.VertexShaderCaps.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexShaderCaps.Equals Method (Object)

Determines whether two [GraphicsDeviceCapabilities.VertexShaderCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [GraphicsDeviceCapabilities.VertexShaderCaps](#) object to compare this instance to.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.VertexShaderCaps](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexShaderCaps left,  
    VertexShaderCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.VertexShaderCaps](#) on the left side of the equal sign.

right

[GraphicsDeviceCapabilities.VertexShaderCaps](#) on the right side of the equal sign.

Return Value

true if *l* is equal to *r*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.VertexShaderCaps.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two [GraphicsDeviceCapabilities.VertexShaderCaps](#) instances are unequal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexShaderCaps left,  
    VertexShaderCaps right  
)
```

Parameters

left

[GraphicsDeviceCapabilities.VertexShaderCaps](#) on the left side of the inequality operator.

right

[GraphicsDeviceCapabilities.VertexShaderCaps](#) on the right side of the inequality operator.

Return Value

true if *l* is not equal to *r*; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsDeviceCapabilities.VertexShaderCaps.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)





[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps Properties

Public Properties

	Name	Description
	DynamicFlowControlDepth	Gets a value specifying the depth of the dynamic flow control instruction nesting.
	NumberTemps	Gets a value specifying the number of temporary registers supported.
	StaticFlowControlDepth	Gets a value specifying the depth of nesting of the loop - vs/rep - vs and call - vs/callnz bool - vs instructions.
	SupportsPredication	Gets a value indicating whether instruction predication is supported.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCapabilities.VertexShaderCaps.DynamicFlowControlDepth Property

Gets a value specifying the depth of the dynamic flow control instruction nesting.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int DynamicFlowControlDepth { get; }
```

Property Value

The depth of the dynamic flow control instruction nesting; either 0 or 24.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.NumberTemps Property

Gets a value specifying the number of temporary registers supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int NumberTemps { get; }
```

Property Value

The number of temporary registers supported.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.StaticFlowControlDepth Property

Gets a value specifying the depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StaticFlowControlDepth { get; }
```

Property Value

The depth of nesting of the **loop - vs/rep - vs** and **call - vs/callnz bool - vs** instructions.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCapabilities.VertexShaderCaps.SupportsPredication Property

Gets a value indicating whether instruction predication is supported.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SupportsPredication { get; }
```

Property Value

true if instruction predication is supported; **false** otherwise.

See Also

Reference

[GraphicsDeviceCapabilities.VertexShaderCaps Structure](#)

[GraphicsDeviceCapabilities.VertexShaderCaps Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCreationParameters Structure

Describes the creation parameters for a device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct GraphicsDeviceCreationParameters
```

See Also

Reference

[GraphicsDeviceCreationParameters Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




GraphicsDeviceCreationParameters Members

The following tables list the members exposed by the GraphicsDeviceCreationParameters type.





Public Constructors

	Name	Description
	GraphicsDeviceCreationParameters	Initializes a new instance of the GraphicsDeviceCreationParameters class.



Public Properties

	Name	Description
	Adapter	Gets the graphics adapter.
	DeviceType	Gets a information describing the amount of emulated functionality for the device.
	FocusWindowHandle	Gets a pointer to the window to which focus belongs for the current device.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCreationParameters Constructor

Initializes a new instance of the [GraphicsDeviceCreationParameters](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDeviceCreationParameters (  
    GraphicsAdapter graphicsAdapter,  
    DeviceType deviceType,  
    IntPtr windowHandle  
)
```

Parameters

graphicsAdapter

The display adapter.

deviceType

The emulated functionality for this device.

windowHandle

Window handle to which focus belongs for this Direct3D device.

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)





[GraphicsDeviceCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCreationParameters Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCreationParameters.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)




[GraphicsDeviceCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCreationParameters Properties

Public Properties

	Name	Description
	Adapter	Gets the graphics adapter.
	DeviceType	Gets a information describing the amount of emulated functionality for the device.
	FocusWindowHandle	Gets a pointer to the window to which focus belongs for the current device.

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsDeviceCreationParameters.Adapter Property

Gets the graphics adapter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsAdapter Adapter { get; }
```

Property Value

The graphics adapter.

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)

[GraphicsDeviceCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCreationParameters.DeviceType Property

Gets a information describing the amount of emulated functionality for the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DeviceType DeviceType { get; }
```

Property Value

The amount of emulated functionality for the device.

RemarksThe value of these parameters mirrors the value passed to the [GraphicsDevice](#) constructor call that created the device.

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)

[GraphicsDeviceCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceCreationParameters.FocusWindowHandle Property

Gets a pointer to the window to which focus belongs for the current device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IntPtr FocusWindowHandle { get; }
```

Property Value

Value that points to the window to which focus belongs for the device.

See Also

Reference

[GraphicsDeviceCreationParameters Structure](#)

[GraphicsDeviceCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsDeviceStatus Enumeration

Describes the status of the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GraphicsDeviceStatus
```

Members

Member name	Description
Lost	The device has been lost.
Normal	The device is normal.
NotReset	The device has not been reset.

See Also

Reference

[GraphicsDevice.GraphicsDeviceStatus Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource Class

Queries and prepares resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class GraphicsResource : IDisposable
```

See Also

Reference

[GraphicsResource Members](#)







[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







GraphicsResource Members

The following tables list the members exposed by the GraphicsResource type.




Public Properties

Name	Description
 GraphicsDevice	Gets the GraphicsDevice associated with this GraphicsResource .
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Name	Gets the name of the resource.
 Priority	Gets or sets the resource-management priority for this resource.
 ResourceType	Gets the type of this resource.
 Tag	Gets the resource tags for this resource.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also







Reference

[GraphicsResource Class](#)




[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsResource Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[GraphicsResource Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsResource.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
GraphicsResource.Dispose ()	Immediately releases the unmanaged resources used by this object.
GraphicsResource.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsResource.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [GraphicsResource](#). The [Dispose](#) method leaves the [GraphicsResource](#) in an unusable state. After calling [Dispose](#), you must release all references to the [GraphicsResource](#) so the garbage collector can reclaim the memory that the [GraphicsResource](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [GraphicsResource](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [GraphicsResource](#) object's [Finalize](#) method.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [GraphicsResource](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the Disposing event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[GraphicsResource Class](#)







[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GraphicsResource Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the GraphicsDevice associated with this GraphicsResource .
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Name	Gets the name of the resource.
	Priority	Gets or sets the resource-management priority for this resource.
	ResourceType	Gets the type of this resource.
	Tag	Gets the resource tags for this resource.

See Also

Reference

[GraphicsResource Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsResource.GraphicsDevice Property

Gets the [GraphicsDevice](#) associated with this [GraphicsResource](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual GraphicsDevice GraphicsDevice { get; }
```

Property Value

The [GraphicsDevice](#) associated with this [GraphicsResource](#).

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.Name Property

Gets the name of the resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of the resource.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.Priority Property

Gets or sets the resource-management priority for this resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Priority { get; set; }
```

Property Value

The new resource-management priority for this resource.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.ResourceType Property

Gets the type of this resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual ResourceType ResourceType { get; }
```

Property Value

The resource type.

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource.Tag Property

Gets the resource tags for this resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tags.

See Also

Reference

[GraphicsResource Class](#)


[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GraphicsResource Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[GraphicsResource Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

GraphicsResource.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[GraphicsResource Class](#)

[GraphicsResource Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceService Interface

Defines a mechanism for retrieving [GraphicsDevice](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public interface IGraphicsDeviceService
```

See Also

Tasks

[Application Model Overview](#)

Reference

[GraphicsDeviceManager](#)

[IGraphicsDeviceService Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





IGraphicsDeviceService Members

The following tables list the members exposed by the IGraphicsDeviceService type.

Public Properties

	Name	Description
	GraphicsDevice	Retrieves a graphics device.

Public Events

	Name	Description
	DeviceCreated	The event that occurs when a graphics device is created.
	DeviceDisposing	The event that occurs when a graphics device is disposing.
	DeviceReset	The event that occurs when a graphics device is reset.
	DeviceResetting	The event that occurs when a graphics device is in the process of resetting.

See Also


Reference

[IGraphicsDeviceService Interface](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IGraphicsDeviceService Properties

Public Properties

	Name	Description
	GraphicsDevice	Retrieves a graphics device.

See Also

Reference

[IGraphicsDeviceService Interface](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IGraphicsDeviceService.GraphicsDevice Property

Retrieves a graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract GraphicsDevice GraphicsDevice { get; }
```

Property Value

A graphics device.

See Also

Reference

[IGraphicsDeviceService Interface](#)





[IGraphicsDeviceService Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceService Events

Public Events

	Name	Description
	DeviceCreated	The event that occurs when a graphics device is created.
	DeviceDisposing	The event that occurs when a graphics device is disposing.
	DeviceReset	The event that occurs when a graphics device is reset.
	DeviceResetting	The event that occurs when a graphics device is in the process of resetting.

See Also

Reference

[IGraphicsDeviceService Interface](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IGraphicsDeviceService.DeviceCreated Event

The event that occurs when a graphics device is created.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract event EventHandler DeviceCreated
```

See Also

Reference

[IGraphicsDeviceService Interface](#)

[IGraphicsDeviceService Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceService.DeviceDisposing Event

The event that occurs when a graphics device is disposing.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract event EventHandler DeviceDisposing
```

See Also

Reference

[IGraphicsDeviceService Interface](#)

[IGraphicsDeviceService Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceService.DeviceReset Event

The event that occurs when a graphics device is reset.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract event EventHandler DeviceReset
```

See Also

Reference

[IGraphicsDeviceService Interface](#)

[IGraphicsDeviceService Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IGraphicsDeviceService.DeviceResetting Event

The event that occurs when a graphics device is in the process of resetting.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract event EventHandler DeviceResetting
```

See Also

Reference

[IGraphicsDeviceService Interface](#)

[IGraphicsDeviceService Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ImageFileFormat Enumeration

Note

This enumeration is available only when developing for Windows.

Defines supported image file formats that may be used for textures.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ImageFileFormat
```

Members

Member name	Description
Bmp	Microsoft Windows bitmap file format.
Dds	DirectDrawSurface file format.
Dib	Microsoft Windows bitmap file format.
Hdr	High dynamic-range file format.
Jpg	Joint Photographic Experts Group (JPEG) compressed file format.
Pfm	Portable float map file format.
Png	Portable Network Graphics file format.
Ppm	Portable pixmap file format.
Tga	Truevision Targa image file format.

See Also

Reference

[Texture.Save Method](#)

[TextureInformation.ImageFormat Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndexBuffer Class

Describes the rendering order of the vertices in a vertex buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class IndexBuffer : GraphicsResource
```

Example

The vertex stream and index data of the graphics device must be set before any call to [DrawIndexedPrimitives](#). The following example sets the index data and associates a user-created vertex buffer of type **VertexPositionNormalTexture** with vertex stream 0 (zero) of the graphics device.

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);  
  
graphics.GraphicsDevice.Indices = lineListIndexBuffer;
```

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[IndexBuffer Members](#)

[Indices](#)

[DrawUserIndexedPrimitives](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista










IndexBuffer Members

The following tables list the members exposed by the IndexBuffer type.









Public Constructors

Name	Description
 IndexBuffer	Overloaded. Initializes a new instance of this class.



Public Properties

Name	Description
 BufferUsage	Gets the state of the related BufferUsage enumeration.
 GraphicsDevice	(Inherited from GraphicsResource .)
 IndexElementSize	Gets or sets a value indicating the size of this index element.
 IsDisposed	(Inherited from GraphicsResource .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 SizeInBytes	Gets the size, in bytes, of this IndexBuffer .
 Tag	(Inherited from GraphicsResource .)


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetData	Overloaded. Copies the index buffer into an array.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetData	Overloaded. Copies array data to the index buffer.
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[IndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IndexBuffer Constructor

Initializes a new instance of this class.

Overload List

Name	Description
IndexBuffer (GraphicsDevice, Int32, BufferUsage, IndexElementSize)	Initializes a new instance of the IndexBuffer class.
IndexBuffer (GraphicsDevice, Type, Int32, BufferUsage)	Initializes a new instance of the IndexBuffer class.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[IndexBuffer](#) Class

[IndexBuffer](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

IndexBuffer Constructor (GraphicsDevice, Int32, BufferUsage, IndexElementSize)

Initializes a new instance of the [IndexBuffer](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IndexBuffer (
    GraphicsDevice graphicsDevice,
    int sizeInBytes,
    BufferUsage usage,
    IndexElementSize elementSize
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) object to associate with the index buffer.

sizeInBytes

The size, in bytes, of the index buffer.

usage

A set of options identifying the behaviors of this index buffer resource.

elementSize

The size, in bits, of an index element.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> <i>elementSize</i> is an invalid value. <i>elementSize</i> must be either IndexElementSize.SixteenBits or IndexElementSize.ThirtyTwoBits. <i>sizeInBytes</i> is not a valid value. <i>sizeInBytes</i> must be greater than zero, and a multiple of <i>elementSize</i>.
InvalidOperationException	Unable to create this IndexBuffer resource on the graphics device.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[IndexBuffer](#) Class

[IndexBuffer](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer Constructor (GraphicsDevice, Type, Int32, BufferUsage)

Initializes a new instance of the [IndexBuffer](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IndexBuffer (
    GraphicsDevice graphicsDevice,
    Type indexType,
    int elementCount,
    BufferUsage usage
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) object to associate with the index buffer.

indexType

The type to use for index values.

elementCount

The number of values in the buffer.

usage

A set of options identifying the behaviors of this index buffer resource.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	The <i>elementCount</i> parameter is invalid. Index buffers can only be created for indices which are sixteen or thirty-two bits in length.
InvalidOperationException	Unable to create this IndexBuffer resource on the graphics device.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[IndexBuffer](#) Class









[IndexBuffer](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace



Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetData	Overloaded. Copies the index buffer into an array.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetData	Overloaded. Copies array data to the index buffer.
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[IndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IndexBuffer.Dispose Method


Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
IndexBuffer.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.
IndexBuffer.Dispose ()	(Inherited from GraphicsResource .)

Remarks

Call [Dispose](#) when you are finished using the [IndexBuffer](#). The [Dispose](#) method leaves the [IndexBuffer](#) in an unusable state. After calling [Dispose](#), you must release all references to the [IndexBuffer](#) so the garbage collector can reclaim the memory that the [IndexBuffer](#) was occupying.

 Note
Always call Dispose before you release your last reference to the IndexBuffer . Otherwise, the resources it is using will not be freed until the garbage collector calls the IndexBuffer object's Finalize method.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IndexBuffer.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the **Finalize** method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. **Finalize** invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [IndexBuffer](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.GetData Method

Copies the index buffer into an array.

Overload List

Name	Description
IndexBuffer.GetData (Int32, T[], Int32, Int32)	Copies the index buffer into an array.
IndexBuffer.GetData (T[])	Copies the index buffer into an array.
IndexBuffer.GetData (T[], Int32, Int32)	Copies the index buffer into an array.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IndexBuffer.GetData Generic Method (Int32, T[], Int32, Int32)

Copies the index buffer into an array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    int offsetInBytes,
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of elements in the array.

Parameters

offsetInBytes

The number of bytes into the index buffer where copying will start.

data

The array to receive index buffer data.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
InvalidOperationException	The array specified in the <i>data</i> parameter is not the correct size for the amount of data requested.
NotSupportedException	This IndexBuffer was created with a usage type of BufferUsage.WriteOnly . Calling GetData on a resource that was created with BufferUsage.WriteOnly is not supported.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.GetData Generic Method (T[])

Copies the index buffer into an array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    T[] data
) where T : ValueType
```

Type Parameters

T

The type of elements in the array.

Parameters

data

The array to receive index buffer data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
InvalidOperationException	The array specified in the <i>data</i> parameter is not the correct size for the amount of data requested.
NotSupportedException	This IndexBuffer was created with a usage type of BufferUsage.WriteOnly . Calling GetData on a resource that was created with BufferUsage.WriteOnly is not supported.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.GetData Generic Method (T[], Int32, Int32)

Copies the index buffer into an array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of elements in the array.

Parameters

data

The array to receive index buffer data.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
InvalidOperationException	The array specified in the <i>data</i> parameter is not the correct size for the amount of data requested.
NotSupportedException	This IndexBuffer was created with a usage type of BufferUsage.WriteOnly . Calling GetData on a resource that was created with BufferUsage.WriteOnly is not supported.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.SetData Method

Copies array data to the index buffer.

Overload List

Name	Description
IndexBuffer.SetData (Int32, T[], Int32, Int32)	Copies array data to the index buffer.
IndexBuffer.SetData (T[])	Copies array data to the index buffer.
IndexBuffer.SetData (T[], Int32, Int32)	Copies array data to the index buffer.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IndexBuffer.SetData Generic Method (Int32, T[], Int32, Int32)

Copies array data to the index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    int offsetInBytes,  
    T[] data,  
    int startIndex,  
    int elementCount  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

offsetInBytes

Number of bytes into the index buffer where copying will start.

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.SetData Generic Method (T[])

Copies array data to the index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    T[] data  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy.

Remarks

⚠Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes. To avoid this potential issue, use [DrawUserPrimitives](#) or [DynamicIndexBuffer](#) as the preferred alternative to **SetData** for dynamic vertex generation.

This is because, in cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Concepts

[Predicated Tiling](#)

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.SetData Generic Method (T[], Int32, Int32)

Copies array data to the index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    T[] data,  
    int startIndex,  
    int elementCount  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[IndexBuffer Class](#)










[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer Properties

Public Properties

	Name	Description
	BufferUsage	Gets the state of the related BufferUsage enumeration.
	GraphicsDevice	(Inherited from GraphicsResource .)
	IndexElementSize	Gets or sets a value indicating the size of this index element.
	IsDisposed	(Inherited from GraphicsResource .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	SizeInBytes	Gets the size, in bytes, of this IndexBuffer .
	Tag	(Inherited from GraphicsResource .)

See Also

Reference

[IndexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

IndexBuffer.BufferUsage Property

Gets the state of the related [BufferUsage](#) enumeration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BufferUsage BufferUsage { get; }
```

Property Value

Indicates how the application uses buffer contents.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.IndexElementSize Property

Gets or sets a value indicating the size of this index element.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IndexElementSize IndexElementSize { get; }
```

Property Value

The size of this index element.

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexBuffer.SizeInBytes Property

Gets the size, in bytes, of this [IndexBuffer](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int SizeInBytes { get; }
```

Property Value

The size, in bytes, of this [IndexBuffer](#).

See Also

Reference

[IndexBuffer Class](#)

[IndexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

IndexElementSize Enumeration

Defines the size of an element of an index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum IndexElementSize
```

Members

Member name	Description
SixteenBits	Sixteen bits.
ThirtyTwoBits	Thirty-two bits.

See Also

Reference

[IndexBuffer Constructor](#)

[IndexBuffer.IndexElementSize Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model Class

Represents a 3D model composed of multiple [ModelMesh](#) objects which may be moved independently.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Model
```

See Also

Tasks

[How To: Render a Model](#)

Reference

[Model Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





Model Members

The following tables list the members exposed by the Model type.









Public Constructors

	Name	Description
	Model	Initializes a new instance of Model .



Public Properties

	Name	Description
	Bones	Gets a collection of ModelBone objects which describe how each mesh in the Meshes collection for this model relates to its parent mesh.
	Meshes	Gets a collection of ModelMesh objects which compose the model. Each ModelMesh in a model may be moved independently and may be composed of multiple materials identified as ModelMeshPart objects.
	Root	Gets the root bone for this model.
	Tag	Gets or sets an object identifying this model.

Public Methods

	Name	Description
	CopyAbsoluteBoneTransformsTo	Copies a transform of each bone in a model relative to all parent bones of the bone into a given array.
	CopyBoneTransformsFrom	Copies an array of transforms into each bone in the model.
	CopyBoneTransformsTo	Copies each bone transform relative only to the parent bone of the model to a given array.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Model Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Model Constructor

Initializes a new instance of [Model](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Model ()
```

See Also

Reference

[Model Class](#)









[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

Model Methods

Public Methods

	Name	Description
	CopyAbsoluteBoneTransformsTo	Copies a transform of each bone in a model relative to all parent bones of the bone into a given array.
	CopyBoneTransformsFrom	Copies an array of transforms into each bone in the model.
	CopyBoneTransformsTo	Copies each bone transform relative only to the parent bone of the model to a given array.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Model Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Model.CopyAbsoluteBoneTransformsTo Method

Copies a transform of each bone in a model relative to all parent bones of the bone into a given array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void CopyAbsoluteBoneTransformsTo (
    Matrix[] destinationBoneTransforms
)
```

Parameters

destinationBoneTransforms

The array to receive bone transforms.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationBoneTransforms</i> is null .
ArgumentOutOfRangeException	<i>destinationBoneTransforms</i> is not the same size as Bones

Remarks

In an absolute transform, each bone is transformed according to the position of all parent bones.

This is the same as iterating the [Bones](#) collection and applying the transformation matrix of every parent of a [ModelBone](#) to the [Transform](#) property of each [ModelBone](#) and copying the results into an array that can be indexed into by the bone index.

An array of transformation matrices for the meshes of any model can be obtained by calling [CopyAbsoluteBoneTransformsTo](#) or [CopyBoneTransformsTo](#). The resulting array contains the transforms that describe how each [ModelMesh](#) is located relative to one another in the [Model](#). The transformation matrix that should be applied to each [ModelMesh](#) can be obtained using the index of the **CopyAbsoluteBoneTransformsTo** to retrieve a transformation matrix from this collection.

For a demonstration of how to use the transforms for the parent bones of a model mesh to set the position of the mesh parts, please see [How To: Render a Model](#).

See Also

Concepts

[What Is a Model Bone?](#)

Tasks

[How To: Render a Model](#)

Reference

[Model Class](#)

[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model.CopyBoneTransformsFrom Method

Copies an array of transforms into each bone in the model.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void CopyBoneTransformsFrom (  
    Matrix[] sourceBoneTransforms  
)
```

Parameters

sourceBoneTransforms

An array containing new bone transforms.

Exceptions

Exception type	Condition
ArgumentNullException	<i>sourceBoneTransforms</i> is null .
ArgumentOutOfRangeException	<i>sourceBoneTransforms</i> is not the same size as Bones

See Also

Concepts

[What Is a Model Bone?](#)

Reference

[Model Class](#)

[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model.CopyBoneTransformsTo Method

Copies each bone transform relative only to the parent bone of the model to a given array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void CopyBoneTransformsTo (
    Matrix[] destinationBoneTransforms
)
```

Parameters

destinationBoneTransforms

The array to receive bone transforms.

Exceptions

Exception type	Condition
ArgumentNullException	<i>destinationBoneTransforms</i> is null .
ArgumentOutOfRangeException	<i>destinationBoneTransforms</i> is not the same size as Bones .

Remarks

This is the same as iterating the [Bones](#) collection and copying the [Transform](#) property into an array that can be indexed into by the bone index.

An array of transformation matrices for the meshes of any model can be obtained by calling [CopyAbsoluteBoneTransformsTo](#) or [CopyBoneTransformsTo](#). The resulting array contains the transforms that describe how each [ModelMesh](#) is located relative to one another in the [Model](#). The transformation matrix that should be applied to each [ModelMesh](#) can be obtained using the index of the [CopyBoneTransformsTo](#) to retrieve a transformation matrix from this collection.

For a demonstration of how to use the transforms for the parent bones of a model mesh to set the position of the mesh parts, please see [How To: Render a Model](#).

See Also

Concepts

[What Is a Model Bone?](#)

Tasks

[How To: Render a Model](#)

Reference

[Model.CopyAbsoluteBoneTransformsTo Method](#)

[Model Class](#)





[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model Properties

Public Properties

	Na me	Description
	Bones	Gets a collection of ModelBone objects which describe how each mesh in the Meshes collection for this model relates to its parent mesh.
	Meshes	Gets a collection of ModelMesh objects which compose the model. Each ModelMesh in a model may be moved independently and may be composed of multiple materials identified as ModelMeshPart objects.
	Root	Gets the root bone for this model.
	Tag	Gets or sets an object identifying this model.

See Also

Reference

[Model Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Model.Bones Property

Gets a collection of [ModelBone](#) objects which describe how each mesh in the [Meshes](#) collection for this model relates to its parent mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBoneCollection Bones { get; }
```

Property Value

A collection of [ModelBone](#) objects used by this model.

Remarks

⚠Caution

Please note that the transforms contained in this collection of bones does not describe the cumulative relationship of each mesh relative to *all* of its parent meshes in the collection, just the immediate parent of the mesh. For this information, please see [CopyAbsoluteBoneTransformsTo](#).

The transforms of each [ModelBone](#) in **Bones** can also be obtained using the [CopyBoneTransformsTo](#) method.

For a demonstration of how to use the transforms for the parent bones of a model mesh to set the position of the mesh parts, please see [How To: Render a Model](#).

See Also

Tasks

[How To: Render a Model](#)

Reference

[Model.CopyAbsoluteBoneTransformsTo Method](#)

[Model Class](#)

[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model.Meshes Property

Gets a collection of [ModelMesh](#) objects which compose the model. Each [ModelMesh](#) in a model may be moved independently and may be composed of multiple materials identified as [ModelMeshPart](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelMeshCollection Meshes { get; }
```

Property Value

A collection of [ModelMesh](#) objects used by this model.

Remarks

For a demonstration of how to use the transforms for the parent bones of a model mesh to set the position of the mesh objects, please see [How To: Render a Model](#).

See Also

Tasks

[How To: Render a Model](#)

Reference

[Model Class](#)

[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model.Root Property

Gets the root bone for this model.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBone Root { get; }
```

Property Value

The root bone for this model.

See Also

Reference

[Model Class](#)

[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Model.Tag Property

Gets or sets an object identifying this model.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

An object identifying this model.

See Also

Reference

[Model Class](#)

[Model Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBone Class

Represents bone data for a model.

Namespace: Microsoft.Xna.Framework.Graphics

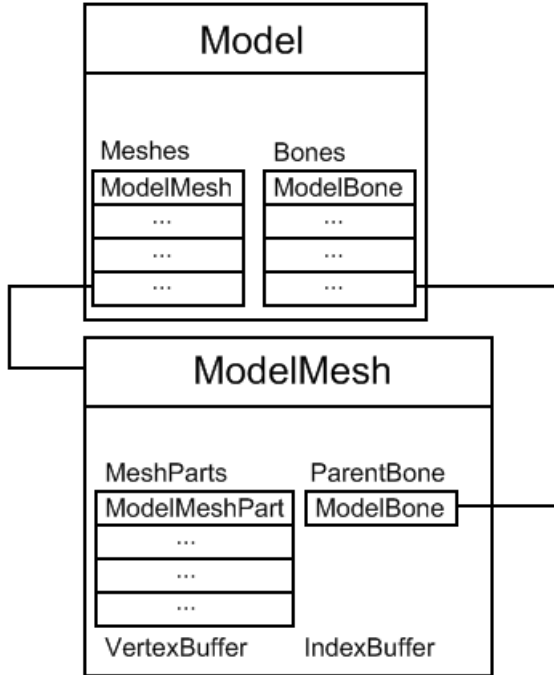
Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelBone
```

Remarks



In the XNA Framework, the [Model](#) class represents the whole model. The [Model](#) contains a [ModelMesh](#) for each separate mesh in the model. Each [ModelMesh](#) contains a [ParentBone](#), which controls the mesh's position and orientation relative to the model. The [Model](#) has a [Root](#) bone, which determines the model's position and orientation. Every [ModelBone](#) can have one parent and many children. The [Root](#) bone on the [Model](#) object is the ultimate parent. Its children are bones on [ModelMesh](#) objects—objects which might have other [ModelMesh](#) bones as their children, and so on. In any given family of bones, rotating the parent bone also rotates the children, and their children, and so on.

Every bone has a transformation matrix (called [Transform](#)) that defines its position and rotation relative to the position of the parent bone. This rotation and translation applies to all the vertices in the [ModelMesh](#) (for example, all the vertices that connect to that bone). To animate a bone, you multiply the default bone transform by a new matrix. When you draw the [ModelMesh](#), you then base your world matrix on the bone's transform.

The easiest way to incorporate transformed bones into drawing is to use the [CopyAbsoluteBoneTransformsTo](#) method. This method takes the bone transforms, which are relative to each other, and iterates over them to make them relative to the [Root](#) bone of the [Model](#). Then it returns a copy of these transforms. When you draw each [ModelMesh](#), you can use the absolute bone transform as the first part of your world matrix. This way you won't have to worry about parent bones and their relationships.

See Also

Concepts

[What Is a Model Bone?](#)

Reference

[ModelBone Members](#)






[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ModelBone Members

The following tables list the members exposed by the ModelBone type.



Public Properties

	Name	Description
	Children	Gets a collection of bones that are children of this bone.
	Index	Gets the index of this bone in the Bones collection.
	Name	Gets the name of this bone.
	Parent	Gets the parent of this bone.
	Transform	Gets or sets the matrix used to transform this bone relative to its parent bone.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelBone Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBone Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelBone Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBone Properties

Public Properties

	Name	Description
	Children	Gets a collection of bones that are children of this bone.
	Index	Gets the index of this bone in the Bones collection.
	Name	Gets the name of this bone.
	Parent	Gets the parent of this bone.
	Transform	Gets or sets the matrix used to transform this bone relative to its parent bone.

See Also

Reference

[ModelBone Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBone.Children Property

Gets a collection of bones that are children of this bone.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBoneCollection Children { get; }
```

Property Value

A collection of bones that are children of this bone.

See Also

Reference

[ModelBone Class](#)

[ModelBone Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBone.Index Property

Gets the index of this bone in the [Bones](#) collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Index { get; }
```

Property Value

The index of this bone in the [Bones](#) collection.

See Also

Reference

[ModelBone Class](#)

[ModelBone Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBone.Name Property

Gets the name of this bone.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this bone.

See Also

Reference

[ModelBone Class](#)

[ModelBone Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBone.Parent Property

Gets the parent of this bone.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBone Parent { get; }
```

Property Value

The parent of this bone.

See Also

Reference

[ModelBone Class](#)

[ModelBone Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBone.Transform Property

Gets or sets the matrix used to transform this bone relative to its parent bone.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix Transform { get; set; }
```

Property Value

The matrix used to transform this bone relative only to its parent bone.

Remarks

⚠Caution

Please note that the transform described by **Transform** does not describe the cumulative relationship of each mesh relative to *all* of its parent meshes in a [Model](#), just the immediate parent of the mesh. For this information, please see [CopyAbsoluteBoneTransformsTo](#).

For a demonstration of how to use the transforms for the parent bones of a model mesh to set the position of the mesh parts, please see [How To: Render a Model](#).

See Also

Tasks

[How To: Render a Model](#)

Reference

[Model.CopyAbsoluteBoneTransformsTo Method](#)

[ModelBone Class](#)

[ModelBone Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection Class

Represents a set of bones associated with a model.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelBoneCollection : ReadOnlyCollection<ModelBone>
```

See Also

Reference

[ModelBoneCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


ModelBoneCollection Members

The following tables list the members exposed by the ModelBoneCollection type.











Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	Overloaded. Retrieves a ModelBone from the collection.



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	Returns a ModelBoneCollection.Enumerator that can iterate through a ModelBoneCollection.
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)
	TryGetValue	Finds a bone with a given name if it exists in the collection.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also











Reference

[ModelBoneCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection .)
	CopyTo	(Inherited from ReadOnlyCollection .)
	Equals	(Inherited from Object .)
	GetEnumerator	Returns a ModelBoneCollection.Enumerator that can iterate through a ModelBoneCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from ReadOnlyCollection .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	TryGetValue	Finds a bone with a given name if it exists in the collection.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ModelBoneCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection.GetEnumerator Method

Returns a [ModelBoneCollection.Enumerator](#) that can iterate through a [ModelBoneCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Enumerator GetEnumerator ()
```

Return Value

The [ModelBoneCollection.Enumerator](#) of the bone collection.

See Also

Reference

[ModelBoneCollection Class](#)

[ModelBoneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection.TryGetValue Method

Finds a bone with a given name if it exists in the collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool TryGetValue (  
    string boneName,  
    out ModelBone value  
)
```

Parameters

boneName

The name of the bone to find.

value

[[OutAttribute](#)] The bone named **boneName**, if found.

Return Value

true if the named bone is found, otherwise **false**.

Exceptions

Exception type	Condition
ArgumentNullException	<i>boneName</i> is null .

See Also

Reference

[ModelBoneCollection Class](#)



[ModelBoneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)


PlatformsXbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	Overloaded. Retrieves a ModelBone from the collection.

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelBoneCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection.Item Property

Retrieves a [ModelBone](#) from the collection.

Overload List

Name	Description
ModelBoneCollection.Item (String)	Retrieves a ModelBone from the collection, given the name of the bone.
ModelBoneCollection.Item (Int32)	(Inherited from ReadOnlyCollection .)

See Also

Reference

[ModelBoneCollection Class](#)

[ModelBoneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection.Item Property (String)

Retrieves a [ModelBone](#) from the collection, given the name of the bone.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBone this [
    string boneName
] { get; }
```

Property Value

The [ModelBone](#) identified by **boneName**.

Exceptions

Exception type	Condition
KeyNotFoundException	<i>boneName</i> was not found in this ModelBoneCollection .

See Also

Reference

[ModelBoneCollection Class](#)

[ModelBoneCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection.Enumerator Structure

Provides the ability to iterate through the bones in an [ModelBoneCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct ModelBoneCollection.Enumerator : IEnumerator<ModelBone>, IDisposable, IEnumer  
ator
```

See Also

Reference

[ModelBoneCollection.Enumerator Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ModelBoneCollection.Enumerator Members

The following tables list the members exposed by the ModelBoneCollection.Enumerator type.



Public Properties

	Name	Description
	Current	Gets the current element in the ModelBoneCollection .



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelBoneCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelBoneCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelBoneCollection .

See Also






Reference

[ModelBoneCollection.Enumerator Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection.Enumerator Methods



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelBoneCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelBoneCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelBoneCollection .

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection.Enumerator.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)

[ModelBoneCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection.Enumerator.MoveNext Method

Advances the enumerator to the next element of the [ModelBoneCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveNext ()
```

Return Value

true if the enumerator was successfully advanced to the next element; **false** if the enumerator has passed the end of the collection.

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

After an enumerator is created or after a call to [System.Collections.IEnumerator.Reset](#), an enumerator is positioned before the first element of the collection, and the first call to **MoveNext** moves the enumerator over the first element of the collection.

After the end of the collection is passed, subsequent calls to **MoveNext** return **false** until [System.Collections.IEnumerator.Reset](#) is called.

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to **MoveNext** or [System.Collections.IEnumerator.Reset](#) throws an [InvalidOperationException](#).

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)

[ModelBoneCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

System.Collections.IEnumerator.Reset Method

Sets the enumerator to its initial position, which is before the first element in the [ModelBoneCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.IEnumerator.Reset ()
```

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to [MoveNext](#) or the **System.Collections.IEnumerator.Reset** throws an [InvalidOperationException](#).

Note

Notes to Implementers: All calls to **System.Collections.IEnumerator.Reset** must result in the same state for the enumerator. The preferred implementation is to move the enumerator to the beginning of the collection, before the first element. This invalidates the enumerator if the collection has been modified since the enumerator was created, which is consistent with [MoveNext](#) and [Current](#).

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)


[ModelBoneCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection.Enumerator Properties

Public Properties

	Name	Description
	Current	Gets the current element in the ModelBoneCollection .

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelBoneCollection.Enumerator.Current Property

Gets the current element in the [ModelBoneCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBone Current { get; }
```

Property Value

The current element in the [ModelBoneCollection](#).

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)

[ModelBoneCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelBoneCollection.Enumerator.System.Collections.IEnumerator.Current Property

Gets the current element in the [ModelBoneCollection](#) as a [Object](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Object System.Collections.IEnumerator.Current { get; }
```

Property Value

The current element in the [ModelBoneCollection](#) as an [Object](#).

See Also

Reference

[ModelBoneCollection.Enumerator Structure](#)

[ModelBoneCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelEffectCollection Class

Represents a collection of effects associated with a model.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelEffectCollection : ReadOnlyCollection<Effect>
```

See Also

Reference

[ModelEffectCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


ModelEffectCollection Members

The following tables list the members exposed by the ModelEffectCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	Returns a ModelEffectCollection.Enumerator that can iterate through a ModelEffectCollection.
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[ModelEffectCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelEffectCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection .)
	CopyTo	(Inherited from ReadOnlyCollection .)
	Equals	(Inherited from Object .)
	GetEnumerator	Returns a ModelEffectCollection.Enumerator that can iterate through a ModelEffectCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from ReadOnlyCollection .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ModelEffectCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelEffectCollection.GetEnumerator Method

Returns a [ModelEffectCollection.Enumerator](#) that can iterate through a [ModelEffectCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Enumerator GetEnumerator ()
```

Return Value

The [ModelEffectCollection.Enumerator](#) of the effect collection.

See Also

Reference

[ModelEffectCollection Class](#)



[ModelEffectCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelEffectCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelEffectCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelEffectCollection.Enumerator Structure

Provides the ability to iterate through the bones in an [ModelEffectCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct ModelEffectCollection.Enumerator : IEnumerator<Effect>, IDisposable, IEnumerator
```

See Also

Reference

[ModelEffectCollection.Enumerator Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ModelEffectCollection.Enumerator Members

The following tables list the members exposed by the ModelEffectCollection.Enumerator type.



Public Properties

Name	Description
 Current	Gets the current element in the ModelEffectCollection .



Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetType	(Inherited from Object .)
 MoveNext	Advances the enumerator to the next element of the ModelEffectCollection .
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerator.Current	Gets the current element in the ModelEffectCollection as a Object .
 System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelEffectCollection .

See Also






Reference

[ModelEffectCollection.Enumerator Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelEffectCollection.Enumerator Methods



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelEffectCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelEffectCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelEffectCollection .

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelEffectCollection.Enumerator.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)

[ModelEffectCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelEffectCollection.Enumerator.MoveNext Method

Advances the enumerator to the next element of the [ModelEffectCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveNext ()
```

Return Value

true if the enumerator was successfully advanced to the next element; **false** if the enumerator has passed the end of the collection.

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

After an enumerator is created or after a call to [System.Collections.IEnumerator.Reset](#), an enumerator is positioned before the first element of the collection, and the first call to **MoveNext** moves the enumerator over the first element of the collection.

After the end of the collection is passed, subsequent calls to **MoveNext** return false until [System.Collections.IEnumerator.Reset](#) is called.

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to **MoveNext** or [System.Collections.IEnumerator.Reset](#) throws an [InvalidOperationException](#).

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)

[ModelEffectCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

System.Collections.IEnumerator.Reset Method

Sets the enumerator to its initial position, which is before the first element in the [ModelEffectCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.IEnumerator.Reset ()
```

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to [MoveNext](#) or the **System.Collections.IEnumerator.Reset** throws an [InvalidOperationException](#).

Note

Notes to Implementers: All calls to **System.Collections.IEnumerator.Reset** must result in the same state for the enumerator. The preferred implementation is to move the enumerator to the beginning of the collection, before the first element. This invalidates the enumerator if the collection has been modified since the enumerator was created, which is consistent with [MoveNext](#) and [Current](#).

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)


[ModelEffectCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelEffectCollection.Enumerator Properties

Public Properties

	Name	Description
	Current	Gets the current element in the ModelEffectCollection .

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelEffectCollection.Enumerator.Current Property

Gets the current element in the [ModelEffectCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Effect Current { get; }
```

Property Value

The current element in the [ModelEffectCollection](#).

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)

[ModelEffectCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelEffectCollection.Enumerator.System.Collections.IEnumerator.Current Property

Gets the current element in the [ModelEffectCollection](#) as a [Object](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Object System.Collections.IEnumerator.Current { get; }
```

Property Value

The current element in the [ModelEffectCollection](#) as a [Object](#).

See Also

Reference

[ModelEffectCollection.Enumerator Structure](#)

[ModelEffectCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh Class

Represents a mesh that is part of a [Model](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelMesh
```

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMesh Members](#)









[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista







ModelMesh Members

The following tables list the members exposed by the ModelMesh type.



Public Properties

Name	Description
 BoundingSphere	Gets the BoundingSphere that contains this mesh.
 Effects	Gets a collection of effects associated with this mesh.
 IndexBuffer	Gets the index buffer for this mesh.
 MeshParts	Gets the ModelMeshPart objects that make up this mesh. Each part of a mesh is composed of a set of primitives that share the same material.
 Name	Gets the name of this mesh.
 ParentBone	Gets the parent bone for this mesh. The parent bone of a mesh contains a transformation matrix that describes how the mesh is located relative to any parent meshes in a model.
 Tag	Gets or sets an object identifying this mesh.
 VertexBuffer	Gets the vertex buffer used to render this mesh.

Public Methods

Name	Description
 Draw	Overloaded. Draws all the ModelMeshPart objects in this mesh, using their current Effect settings.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also







Reference

[ModelMesh Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMesh Methods

Public Methods

	Name	Description
	Draw	Overloaded. Draws all the ModelMeshPart objects in this mesh, using their current Effect settings.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ModelMesh Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMesh.Draw Method

Draws all the [ModelMeshPart](#) objects in this mesh, using their current [Effect](#) settings.

Overload List

Name	Description
ModelMesh.Draw ()	Draws all of the ModelMeshPart objects in this mesh, using their current Effect settings.
ModelMesh.Draw (SaveStateMode)	Draws all of the ModelMeshPart objects in this mesh, using their current Effect settings, and specifying options for saving effect state.

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMesh.Draw Method ()

Draws all of the [ModelMeshPart](#) objects in this mesh, using their current [Effect](#) settings.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Effect is a null .

Remarks

Unexpected behavior can result when rendering sprites and 3D objects on the same graphics device. For more information, see [Render States and the Graphics Device](#).

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.Draw Method (SaveStateMode)

Draws all of the [ModelMeshPart](#) objects in this mesh, using their current [Effect](#) settings, and specifying options for saving effect state.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (  
    SaveStateMode saveStateMode  
)
```

Parameters

saveStateMode

The save state options to pass to each [Effect](#).

Exceptions

Exception type	Condition
InvalidOperationException	Effect is a null .

Remarks

Unexpected behavior can result when rendering sprites and 3D objects on the same graphics device. For more information, see [Render States and the Graphics Device](#).

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMesh Class](#)









[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh Properties

Public Properties

	Name	Description
	BoundingSphere	Gets the BoundingSphere that contains this mesh.
	Effects	Gets a collection of effects associated with this mesh.
	IndexBuffer	Gets the index buffer for this mesh.
	MeshParts	Gets the ModelMeshPart objects that make up this mesh. Each part of a mesh is composed of a set of primitives that share the same material.
	Name	Gets the name of this mesh.
	ParentBone	Gets the parent bone for this mesh. The parent bone of a mesh contains a transformation matrix that describes how the mesh is located relative to any parent meshes in a model.
	Tag	Gets or sets an object identifying this mesh.
	VertexBuffer	Gets the vertex buffer used to render this mesh.

See Also

Reference

[ModelMesh Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMesh.BoundingBoxSphere Property

Gets the [BoundingBoxSphere](#) that contains this mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BoundingBoxSphere BoundingBoxSphere { get; }
```

Property Value

The [BoundingBoxSphere](#) that contains this mesh.

See Also

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.Effects Property

Gets a collection of effects associated with this mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelEffectCollection Effects { get; }
```

Property Value

A collection of effects associated with this mesh.

Remarks

[ModelMesh.Effects](#) is a shortcut for iterating over the [MeshParts](#) property of the [ModelMesh](#) and looking up the [ModelMeshPart.Effect](#) property of each [ModelMeshPart](#).

[ModelMesh.Effects](#) is a collection of all the [Effect](#) properties of the for the [MeshParts](#) of this [ModelMesh](#). Each [ModelMeshPart](#) has a single [Effect](#) which is a reference to one of the [Effects](#) of the parent [ModelMesh](#) property. By updating all the effects of the [ModelMesh](#) all of the effects of each [ModelMeshPart](#) are updated as well.

See Also

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.IndexBuffer Property

Gets the index buffer for this mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IndexBuffer IndexBuffer { get; }
```

Property Value

The index buffer for this mesh.

See Also

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.MeshParts Property

Gets the [ModelMeshPart](#) objects that make up this mesh. Each part of a mesh is composed of a set of primitives that share the same material.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelMeshPartCollection MeshParts { get; }
```

Property Value

The [ModelMeshPart](#) objects that make up this mesh.

Remarks

It is not necessary to use this class directly. In advanced rendering scenarios, it is possible to draw using **MeshParts** properties in combination with the vertex and index buffers on [ModelMesh](#). However, in most cases, [ModelMesh.Draw](#) will be sufficient.

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.Name Property

Gets the name of this mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this mesh.

See Also

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.ParentBone Property

Gets the parent bone for this mesh. The parent bone of a mesh contains a transformation matrix that describes how the mesh is located relative to any parent meshes in a model.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelBone ParentBone { get; }
```

Property Value

The parent bone for this mesh.

Remarks

An array of transformation matrices for the meshes of any model can be obtained by calling [CopyAbsoluteBoneTransformsTo](#) or [CopyBoneTransformsTo](#). The resulting array contains the transforms that describe how each [ModelMesh](#) is located relative to one another in the [Model](#). The transformation matrix that should be applied to each [ModelMesh](#) can be obtained using the index of the **ParentBone** to retrieve a transformation matrix from this collection.

For a demonstration of how to use the transforms for the parent bones of a model mesh to set the position of the mesh parts, please see [How To: Render a Model](#).

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.Tag Property

Gets or sets an object identifying this mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

An object identifying this mesh.

See Also

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMesh.VertexBuffer Property

Gets the vertex buffer used to render this mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexBuffer VertexBuffer { get; }
```

Property Value

The vertex buffer used to render this mesh.

See Also

Reference

[ModelMesh Class](#)

[ModelMesh Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection Class

Represents a collection of [ModelMesh](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelMeshCollection : ReadOnlyCollection<ModelMesh>
```

See Also

Reference

[ModelMeshCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


ModelMeshCollection Members

The following tables list the members exposed by the ModelMeshCollection type.











Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	Overloaded. Retrieves a ModelMesh from the collection.



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	Returns a ModelMeshCollection.Enumerator that can iterate through a ModelMeshCollection.
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)
	TryGetValue	Finds a mesh with a given name if it exists in the collection.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also











Reference

[ModelMeshCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	Returns a ModelMeshCollection.Enumerator that can iterate through a ModelMeshCollection.
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)
	TryGetValue	Finds a mesh with a given name if it exists in the collection.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also

Reference

[ModelMeshCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection.GetEnumerator Method

Returns a [ModelMeshCollection.Enumerator](#) that can iterate through a [ModelMeshCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Enumerator GetEnumerator ()
```

Return Value

The [ModelMeshCollection.Enumerator](#) of the mesh collection.

See Also

Reference

[ModelMeshCollection Class](#)

[ModelMeshCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection.TryGetValue Method

Finds a mesh with a given name if it exists in the collection.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool TryGetValue (  
    string meshName,  
    out ModelMesh value  
)
```

Parameters

meshName

The name of the mesh to find.

value

[[OutAttribute](#)] The mesh named **meshName**, if found.

Return Value

true if the named mesh is found; **false** otherwise.

Exceptions

Exception type	Condition
ArgumentNullException	<i>meshName</i> is null .

See Also

Reference

[ModelMeshCollection Class](#)



[ModelMeshCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	Overloaded. Retrieves a ModelMesh from the collection.

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelMeshCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection.Item Property

Retrieves a [ModelMesh](#) from the collection.

Overload List

Name	Description
ModelMeshCollection.Item (String)	Retrieves a ModelMesh from the collection, given the name of the mesh.
ModelMeshCollection.Item (Int32)	(Inherited from ReadOnlyCollection .)

See Also

Reference

[ModelMeshCollection Class](#)

[ModelMeshCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection.Item Property (String)

Retrieves a [ModelMesh](#) from the collection, given the name of the mesh.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelMesh this [
    string meshName
] { get; }
```

Property Value

The [ModelMesh](#) identified by **meshName**.

Exceptions

Exception type	Condition
KeyNotFoundException	<i>meshName</i> was not found in this ModelMeshCollection .

See Also

Reference

[ModelMeshCollection Class](#)

[ModelMeshCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection.Enumerator Structure

Provides the ability to iterate through the bones in an [ModelMeshCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct ModelMeshCollection.Enumerator : IEnumerator<ModelMesh>, IDisposable, IEnumer  
ator
```

See Also

Reference

[ModelMeshCollection.Enumerator Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ModelMeshCollection.Enumerator Members

The following tables list the members exposed by the ModelMeshCollection.Enumerator type.



Public Properties

	Name	Description
	Current	Gets the current element in the ModelMeshCollection .



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelMeshCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelMeshCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelMeshCollection .






See Also

Reference



[ModelMeshCollection.Enumerator Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection.Enumerator Methods



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelMeshCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelMeshCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelMeshCollection .

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection.Enumerator.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)

[ModelMeshCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection.Enumerator.MoveNext Method

Advances the enumerator to the next element of the [ModelMeshCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveNext ()
```

Return Value

true if the enumerator was successfully advanced to the next element; **false** if the enumerator has passed the end of the collection.

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

After an enumerator is created or after a call to [System.Collections.IEnumerator.Reset](#), an enumerator is positioned before the first element of the collection, and the first call to **MoveNext** moves the enumerator over the first element of the collection.

After the end of the collection is passed, subsequent calls to **MoveNext** return false until [System.Collections.IEnumerator.Reset](#) is called.

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to **MoveNext** or [System.Collections.IEnumerator.Reset](#) throws an [InvalidOperationException](#).

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)

[ModelMeshCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

System.Collections.IEnumerator.Reset Method

Sets the enumerator to its initial position, which is before the first element in the [ModelMeshCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.IEnumerator.Reset ()
```

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to [MoveNext](#) or the **System.Collections.IEnumerator.Reset** throws an [InvalidOperationException](#).

Note

Notes to Implementers: All calls to **System.Collections.IEnumerator.Reset** must result in the same state for the enumerator. The preferred implementation is to move the enumerator to the beginning of the collection, before the first element. This invalidates the enumerator if the collection has been modified since the enumerator was created, which is consistent with [MoveNext](#) and [Current](#).

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)


[ModelMeshCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection.Enumerator Properties

Public Properties

	Name	Description
	Current	Gets the current element in the ModelMeshCollection .

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshCollection.Enumerator.Current Property

Gets the current element in the [ModelMeshCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelMesh Current { get; }
```

Property Value

The current element in the [ModelMeshCollection](#).

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)

[ModelMeshCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshCollection.Enumerator.System.Collections.IEnumerator.Current Property

Gets the current element in the [ModelMeshCollection](#) as a [Object](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Object System.Collections.IEnumerator.Current { get; }
```

Property Value

The current element in the [ModelMeshCollection](#).

See Also

Reference

[ModelMeshCollection.Enumerator Structure](#)

[ModelMeshCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart Class

Represents a batch of geometry information to submit to the graphics device during rendering. Each **ModelMeshPart** is a subdivision of a [ModelMesh](#) object. The [ModelMesh](#) class is split into multiple **ModelMeshPart** objects, typically based on material information.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelMeshPart
```

Remarks

It is not necessary to use this class directly. In advanced rendering scenarios, it is possible to draw using **ModelMeshPart** properties in combination with the vertex and index buffers on [ModelMesh](#). However, in most cases, [ModelMesh.Draw](#) will be sufficient.

See Also

Tasks

[How To: Render a Model](#)

Reference

[ModelMeshPart Members](#)










[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ModelMeshPart Members

The following tables list the members exposed by the ModelMeshPart type.



Public Properties

Name	Description
 BaseVertex	Gets the offset to add to each vertex index in the index buffer.
 Effect	Gets or sets the material Effect for this mesh part.
 NumVertices	Gets the number of vertices used during a draw call.
 PrimitiveCount	Gets the number of primitives to render.
 StartIndex	Gets the location in the index array at which to start reading vertices.
 StreamOffset	Gets the offset in bytes from the beginning of the VertexBuffer .
 Tag	Gets or sets an object identifying this model mesh part.
 VertexDeclaration	Gets the vertex declaration for this model mesh part.
 VertexStride	Gets the size, in bytes, of the elements in this vertex stream.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also










Reference

[ModelMeshPart Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPart Properties

Public Properties

	Name	Description
	BaseVertex	Gets the offset to add to each vertex index in the index buffer.
	Effect	Gets or sets the material Effect for this mesh part.
	NumVertices	Gets the number of vertices used during a draw call.
	PrimitiveCount	Gets the number of primitives to render.
	StartIndex	Gets the location in the index array at which to start reading vertices.
	StreamOffset	Gets the offset in bytes from the beginning of the VertexBuffer .
	Tag	Gets or sets an object identifying this model mesh part.
	VertexDeclaration	Gets the vertex declaration for this model mesh part.
	VertexStride	Gets the size, in bytes, of the elements in this vertex stream.

See Also

Reference

[ModelMeshPart Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPart.BaseVertex Property

Gets the offset to add to each vertex index in the index buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BaseVertex { get; }
```

Property Value

Offset to add to each vertex index in the index buffer.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.Effect Property

Gets or sets the material [Effect](#) for this mesh part.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Effect Effect { get; set; }
```

Property Value

The material effect for this mesh part.

Remarks

[ModelMesh.Effects](#) is a shortcut for iterating over the [MeshParts](#) property of the [ModelMesh](#) and looking up the [ModelMeshPart.Effect](#) property of each [ModelMeshPart](#).

[ModelMesh.Effects](#) is a collection of all the [Effect](#) properties of the for the [MeshParts](#) of this [ModelMesh](#). Each [ModelMeshPart](#) has a single [Effect](#) which is a reference to one of the [Effects](#) of the parent [ModelMesh](#) property. By updating all the effects of the [ModelMesh](#) all of the effects of each [ModelMeshPart](#) are updated as well.

Example

The **Effect** property can be used to assign a custom [Effect](#) to a model loaded with [BasicEffect](#).

C#

```
public static void RemapModel(Model model, Effect effect)
{
    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (ModelMeshPart part in mesh.MeshParts)
        {
            part.Effect = effect;
        }
    }
}
```

See Also

Tasks

[How To: Render a Model](#)

[How To: Draw a Model with a Custom Effect](#)

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.NumVertices Property

Gets the number of vertices used during a draw call.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int NumVertices { get; }
```

Property Value

The number of vertices used during the call.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.PrimitiveCount Property

Gets the number of primitives to render.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PrimitiveCount { get; }
```

Property Value

The number of primitives to render. The number of vertices used is a function of primitiveCount and primitiveType.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.StartIndex Property

Gets the location in the index array at which to start reading vertices.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StartIndex { get; }
```

Property Value

Location in the index array at which to start reading vertices.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.StreamOffset Property

Gets the offset in bytes from the beginning of the [VertexBuffer](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StreamOffset { get; }
```

Property Value

The offset in bytes from the beginning of the [VertexBuffer](#).

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.Tag Property

Gets or sets an object identifying this model mesh part.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

An object identifying this model mesh part.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.VertexDeclaration Property

Gets the vertex declaration for this model mesh part.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexDeclaration VertexDeclaration { get; }
```

Property Value

The vertex declaration for this model mesh part.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPart.VertexStride Property

Gets the size, in bytes, of the elements in this vertex stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int VertexStride { get; }
```

Property Value

The size, in bytes, of the elements in this vertex stream.

See Also

Reference

[ModelMeshPart Class](#)

[ModelMeshPart Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPartCollection Class

Represents a collection of [ModelMeshPart](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ModelMeshPartCollection : ReadOnlyCollection<ModelMeshPart>
```

See Also

Reference

[ModelMeshPartCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista


ModelMeshPartCollection Members

The following tables list the members exposed by the ModelMeshPartCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	Returns a ModelMeshPartCollection.Enumerator that can iterate through a ModelMeshPartCollection.
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[ModelMeshPartCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPartCollection Methods

Public Methods

Name	Description
 Contains	(Inherited from ReadOnlyCollection .)
 CopyTo	(Inherited from ReadOnlyCollection .)
 Equals	(Inherited from Object .)
 GetEnumerator	Returns a ModelMeshPartCollection.Enumerator that can iterate through a ModelMeshPartCollection .
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from ReadOnlyCollection .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ModelMeshPartCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPartCollection.GetEnumerator Method

Returns a [ModelMeshPartCollection.Enumerator](#) that can iterate through a [ModelMeshPartCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Enumerator GetEnumerator ()
```

Return Value

The [ModelMeshPartCollection.Enumerator](#) of the [ModelMeshPart](#) collection.

See Also

Reference

[ModelMeshPartCollection Class](#)



[ModelMeshPartCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPartCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelMeshPartCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPartCollection.Enumerator Structure

Provides the ability to iterate through the bones in an [ModelMeshPartCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct ModelMeshPartCollection.Enumerator : IEnumerator<ModelMeshPart>, IDisposable,
IEnumerator
```

See Also

Reference

[ModelMeshPartCollection.Enumerator Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






ModelMeshPartCollection.Enumerator Members

The following tables list the members exposed by the ModelMeshPartCollection.Enumerator type.



Public Properties

	Name	Description
	Current	Gets the current element in the ModelMeshPartCollection .



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelMeshPartCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelMeshPartCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelMeshPartCollection .






See Also

Reference



[ModelMeshPartCollection.Enumerator Structure](#)
[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPartCollection.Enumerator Methods



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the ModelMeshPartCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the ModelMeshPartCollection as a Object .
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the ModelMeshPartCollection .

See Also

Reference

[ModelMeshPartCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPartCollection.Enumerator.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ModelMeshPartCollection.Enumerator Structure](#)

[ModelMeshPartCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPartCollection.Enumerator.MoveNext Method

Advances the enumerator to the next element of the [ModelMeshPartCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveNext ()
```

Return Value

true if the enumerator was successfully advanced to the next element; **false** if the enumerator has passed the end of the collection.

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

After an enumerator is created or after a call to [System.Collections.IEnumerator.Reset](#), an enumerator is positioned before the first element of the collection, and the first call to **MoveNext** moves the enumerator over the first element of the collection.

After the end of the collection is passed, subsequent calls to **MoveNext** return false until [System.Collections.IEnumerator.Reset](#) is called.

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to **MoveNext** or [System.Collections.IEnumerator.Reset](#) throws an [InvalidOperationException](#).

See Also

Reference

[ModelMeshPartCollection.Enumerator Structure](#)

[ModelMeshPartCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

System.Collections.IEnumerator.Reset Method

Sets the enumerator to its initial position, which is before the first element in the [ModelMeshPartCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.IEnumerator.Reset ()
```

Exceptions

Exception type	Condition
InvalidOperationException	The collection was modified after the enumerator was created.

Remarks

An enumerator remains valid as long as the collection remains unchanged. If changes, such as adding, modifying, or deleting elements are made to the collection, the enumerator is irrecoverably invalidated and the next call to [MoveNext](#) or the **System.Collections.IEnumerator.Reset** throws an [InvalidOperationException](#).

Note

Notes to Implementers: All calls to **System.Collections.IEnumerator.Reset** must result in the same state for the enumerator. The preferred implementation is to move the enumerator to the beginning of the collection, before the first element. This invalidates the enumerator if the collection has been modified since the enumerator was created, which is consistent with [MoveNext](#) and [Current](#).

See Also

Reference

[ModelMeshPartCollection.Enumerator Structure](#)


[ModelMeshPartCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPartCollection.Enumerator Properties

Public Properties

	Name	Description
	Current	Gets the current element in the ModelMeshPartCollection .

See Also

Reference

[ModelMeshPartCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ModelMeshPartCollection.Enumerator.Current Property

Gets the current element in the [ModelMeshPartCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ModelMeshPart Current { get; }
```

Property Value

The current element in the [ModelMeshPartCollection](#).

See Also

Reference

[ModelMeshPartCollection.Enumerator Structure](#)

[ModelMeshPartCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ModelMeshPartCollection.Enumerator.System.Collections.IEnumerator.Current Property

Gets the current element in the [ModelMeshPartCollection](#) as a [Object](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Object System.Collections.IEnumerator.Current { get; }
```

Property Value

The current element in the [ModelBoneCollection](#) as an [Object](#).

See Also

Reference

[ModelMeshPartCollection.Enumerator](#) Structure

[ModelMeshPartCollection.Enumerator](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista

MultiSampleType Enumeration

Defines the levels of full-scene multisampling that the game machine can apply.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum MultiSampleType
```

Members

Member name	Description
SixteenSamples	Enables 16 levels of full-scene multisampling.
FifteenSamples	Enables 15 levels of full-scene multisampling.
FourteenSamples	Enables 14 levels of full-scene multisampling.
ThirteenSamples	Enables 13 levels of full-scene multisampling.
TwelveSamples	Enables 12 levels of full-scene multisampling.
ElevenSamples	Enables 11 levels of full-scene multisampling.
TenSamples	Enables 10 levels of full-scene multisampling.
NineSamples	Enables 9 levels of full-scene multisampling.
EightSamples	Enables 8 levels of full-scene multisampling.
SevenSamples	Enables 7 levels of full-scene multisampling.
SixSamples	Enables 6 levels of full-scene multisampling.
FiveSamples	Enables 5 levels of full-scene multisampling.
FourSamples	Enables 4 levels of full-scene multisampling.
ThreeSamples	Enables 3 levels of full-scene multisampling.
TwoSamples	Enables 2 levels of full-scene multisampling.
NonMaskable	Enables the multisample quality value. For a quality of 1, the video card determines the best multisampling mode.
None	Specifies no full-scene multisampling, and enables swap effects other than SwapEffect.Discard .

Remarks

When multisampling is enabled, all subsamples of a pixel are updated in one pass. However, when subsamples are used for other effects that involve multiple rendering passes, the application can specify that only some of them be affected by a given rendering pass. This latter approach enables simulation of motion blur, depth-of-field focus effects, reflection blur, and other effects.

In addition to enabling full-scene multisampling when a [Reset](#) occurs, there are render states that turn various multisampling aspects on and off at fine-grained levels. For more information, see [RenderState](#).

Multisampling is valid only on a swap chain that is created or reset with [SwapEffect.Discard](#).

The multisampling antialiasing value can be set with the parameters or sub-parameters in the following methods.

Module	Parameters	Sub-parameters
CheckDeviceMultiSampleType	<i>multiSampleType, qualityLevels</i>	None
GraphicsDevice	<i>presentationParameters</i>	MultiSampleType , MultiSampleQuality
Reset	<i>presentationParameters</i>	MultiSampleType , MultiSampleQuality

Example

Use the [PreparingDeviceSettings](#) event to set **MultiSampleType** after calling [CheckDeviceMultiSampleType](#):

C#

```
void graphics_PreparingDeviceSettings(object sender,
    PreparingDeviceSettingsEventArgs e)
{
    // Xbox 360 and most PCs support FourSamples/0
```

```

        // (4x) and TwoSamples/0 (2x) antialiasing.
        PresentationParameters pp =
            e.GraphicsDeviceInformation.PresentationParameters;
    #if XBOX
        pp.MultiSampleQuality = 0;
        pp.MultiSampleType = MultiSampleType.FourSamples;
        return;
    #else
        int quality = 0;
        GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
        SurfaceFormat format = adapter.CurrentDisplayMode.Format;
        // Check for 4xAA
        if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
            false, MultiSampleType.FourSamples, out quality))
        {
            // even if a greater quality is returned, we only want quality 0
            pp.MultiSampleQuality = 0;
            pp.MultiSampleType =
                MultiSampleType.FourSamples;
        }
        // Check for 2xAA
        else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
            format, false, MultiSampleType.TwoSamples, out quality))
        {
            // even if a greater quality is returned, we only want quality 0
            pp.MultiSampleQuality = 0;
            pp.MultiSampleType =
                MultiSampleType.TwoSamples;
        }
        return;
    #endif
}

```

See Also

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[GraphicsAdapter.CheckDeviceMultiSampleType Method](#)

[PresentationParameters.MultiSampleQuality Property](#)

[PresentationParameters.MultiSampleType Property](#)

[DepthStencilBuffer.MultiSampleType Property](#)

[SwapEffect Enumeration](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

OcclusionQuery Class

Used to perform an occlusion query against the latest drawn objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class OcclusionQuery : IDisposable
```

Remarks

An occlusion query is a technique that determines how many pixels were actually drawn during a set of draw calls. This is useful for certain rendering techniques and rendering optimizations.

An example of using this query follows:

```
OcclusionQuery query = new OcclusionQuery(device);

query.Begin();
// Drawing simple objects, bounding areas, etc.
query.End();

// Do additional work here to provide enough time for the GPU to complete query execution.

// Draw additional models
if (query.IsComplete == true && query.PixelCount > 0)
{
    // A non-zero pixel count means some of the low res model is visible
    // so let's draw the real version of it
}
```

The number of non-occluded pixels (indicated by [PixelCount](#)) can differ based on the platform.

For Xbox 360 games, [PixelCount](#) represents the number of pixels modified by drawing. This includes [GraphicsDevice.Clear](#) calls and any drawing associated with sprite batches.

For Windows games, [PixelCount](#) represents the number of pixels that passed the depth and stencil tests. This does not include sprite batch drawing (because the depth test is turned off) or calls to [GraphicsDevice.Clear](#).

To achieve consistent results across all platforms, use occlusion queries only when the depth buffer is enabled. In addition, avoid calling [Clear](#) or changing the assigned render target inside an occlusion query begin/end block.

See Also

Reference

[OcclusionQuery Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








OcclusionQuery Members

The following tables list the members exposed by the OcclusionQuery type.









Public Constructors

Name	Description
 OcclusionQuery	Initializes a new instance of OcclusionQuery with the specified device.




Public Properties

Name	Description
 GraphicsDevice	Gets the graphics device being queried.
 IsComplete	Gets a value that indicates if the occlusion query has completed.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 IsSupported	Specifies if occlusion queries are supported on the related graphics device.
 Name	Gets the name of this occlusion query
 PixelCount	Gets the number of visible pixels.
 Tag	Gets the resource tag for this occlusion query.


Public Methods

Name	Description
 Begin	Begins application of the query.
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 End	Ends the application of the query.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[OcclusionQuery Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OcclusionQuery Constructor

Initializes a new instance of **OcclusionQuery** with the specified device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public OcclusionQuery (  
    GraphicsDevice graphicsDevice  
)
```

Parameters

graphicsDevice

The graphics device to associate with this query.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
InvalidOperationException	This resource could not be created.

See Also

Reference

[OcclusionQuery Class](#)









[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery Methods

Public Methods

	Name	Description
	Begin	Begins application of the query.
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	End	Ends the application of the query.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[OcclusionQuery Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OcclusionQuery.Begin Method

Begins application of the query.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin ()
```

Exceptions

Exception type	Condition
InvalidOperationException	<ul style="list-style-type: none">• Begin cannot be called again until End has been successfully called.• Begin may not be called on this query object again before IsComplete has been checked.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
OcclusionQuery.Dispose ()	Immediately releases the unmanaged resources used by this object.
OcclusionQuery.Dispose (Boolean)	Releases the unmanaged resources used by Dispose and optionally releases the managed resources.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OcclusionQuery.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.Dispose Method (Boolean)

Releases the unmanaged resources used by **Dispose** and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.End Method

Ends the application of the query.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void End ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Begin must be called successfully before End can be called.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OcclusionQuery.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[OcclusionQuery Class](#)








[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the graphics device being queried.
	IsComplete	Gets a value that indicates if the occlusion query has completed.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	IsSupported	Specifies if occlusion queries are supported on the related graphics device.
	Name	Gets the name of this occlusion query
	PixelCount	Gets the number of visible pixels.
	Tag	Gets the resource tag for this occlusion query.

See Also

Reference

[OcclusionQuery Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OcclusionQuery.GraphicsDevice Property

Gets the graphics device being queried.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The related graphics device queried the occlusion query.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.IsComplete Property

Gets a value that indicates if the occlusion query has completed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsComplete { get; }
```

Property Value

true if the query completed; otherwise **false**.

Remarks

If predicated tiling has been triggered for an Xbox 360 game, calls to **IsComplete** from within the tiling bracket always fail. Access this data by switching to a different render target or end the frame. This condition does not apply to games targeting the Windows platform.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; otherwise **false**.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.IsSupported Property

Specifies if occlusion queries are supported on the related graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsSupported { get; }
```

Property Value

true if occlusion queries are supported; **false** otherwise.

Occlusion queries are always supported on the Xbox 360 platform. Occlusion queries on the Windows platform are dependent upon the abilities of the installed video card.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.Name Property

Gets the name of this occlusion query

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

Name of the query.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.PixelCount Property

Gets the number of visible pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PixelCount { get; }
```

Property Value

Number of visible pixels. Zero indicates full occlusion, which means the pixels are not visible from the current camera position.

Exceptions

Exception type	Condition
InvalidOperationException	The query data is not yet available. Use the IsComplete property to determine if the data is available before attempting to retrieve it.

Remarks

The number of non-occluded pixels (indicated by [PixelCount](#)) can differ based on the platform.

For Xbox 360 games, [PixelCount](#) represents the number of pixels modified by drawing. This includes [GraphicsDevice.Clear](#) calls and any drawing associated with sprite batches.

For Windows games, [PixelCount](#) represents the number of pixels that passed the depth and stencil tests. This does not include sprite batch drawing (because the depth test is turned off) or calls to [GraphicsDevice.Clear](#).

To achieve consistent results across all platforms, use occlusion queries only when the depth buffer is enabled. In addition, avoid calling [Clear](#) or changing the assigned render target inside an occlusion query begin/end block.

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery.Tag Property

Gets the resource tag for this occlusion query.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tag for this query.

See Also

Reference

[OcclusionQuery Class](#)


[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OcclusionQuery Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[OcclusionQuery Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OcclusionQuery.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[OcclusionQuery Class](#)

[OcclusionQuery Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

OutOfVideoMemoryException Class

The exception that is thrown when Direct3D does not have enough display memory to perform the operation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class OutOfVideoMemoryException : ExternalException
```

See Also

Reference

[OutOfVideoMemoryException Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









OutOfVideoMemoryException Members

The following tables list the members exposed by the OutOfVideoMemoryException type.


Public Constructors

Name	Description
 OutOfVideoMemoryException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[OutOfVideoMemoryException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OutOfVideoMemoryException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
OutOfVideoMemoryException ()	Initializes a new instance of this class.
OutOfVideoMemoryException (String)	Initializes a new instance of this class with a specified error message.
OutOfVideoMemoryException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[OutOfVideoMemoryException Class](#)

[OutOfVideoMemoryException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

OutOfVideoMemoryException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public OutOfVideoMemoryException ()
```

Remarks

This constructor initializes the [Message](#) property of the new instance to a system-supplied message that describes the error, such as "Direct3D does not have enough display memory to perform the operation." This message takes into account the current system culture.

The following table shows the initial property values for an instance of [OutOfVideoMemoryException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[OutOfVideoMemoryException Class](#)

[OutOfVideoMemoryException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

OutOfVideoMemoryException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public OutOfVideoMemoryException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [OutOfVideoMemoryException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[OutOfVideoMemoryException Class](#)

[OutOfVideoMemoryException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

OutOfVideoMemoryException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public OutOfVideoMemoryException (  
    string message,  
    Exception inner  
)
```

Parameters

message

A message that describes the error.

inner

The exception that is the cause of the current exception. If the *inner* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [OutOfVideoMemoryException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[OutOfVideoMemoryException Class](#)







[OutOfVideoMemoryException Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

OutOfVideoMemoryException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[OutOfVideoMemoryException Class](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

OutOfVideoMemoryException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[OutOfVideoMemoryException Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PixelShader Class

Encapsulates the functionality of a pixel shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class PixelShader : IDisposable
```

See Also

Reference

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





PixelShader Members

The following tables list the members exposed by the PixelShader type.








Public Constructors

Name	Description
 PixelShader	Initializes a new instance of the PixelShader class.




Public Properties

Name	Description
 GraphicsDevice	Gets the GraphicsDevice associated with this PixelShader .
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Name	Returns the name of the pixel shader.
 Tag	Gets the resource tags.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetShaderCode	Gets an array of bytes containing the shader code.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[PixelShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PixelShader Constructor

Initializes a new instance of the [PixelShader](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PixelShader (  
    GraphicsDevice graphicsDevice,  
    byte[] shaderCode  
)
```

Parameters

graphicsDevice

Device to create the pixel shader.

shaderCode

The compiled byte code. A pixel shader function token array, specifying the blending operations.

Exceptions

Exception type	Condition
ArgumentException	The array specified in <i>shaderCode</i> must have a length that is a multiple of four.
ArgumentNullException	<i>graphicsDevice</i> or <i>shaderCode</i> is null .
InvalidOperationException	Unable to create this PixelShader resource on the graphics device.

See Also

Reference

[PixelShader Class](#)








[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetShaderCode	Gets an array of bytes containing the shader code.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[PixelShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PixelShader.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
PixelShader.Dispose ()	Immediately releases the unmanaged resources used by this object.
PixelShader.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PixelShader.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [PixelShader](#). The [Dispose](#) method leaves the [PixelShader](#) in an unusable state. After calling [Dispose](#), you must release all references to the [PixelShader](#) so the garbage collector can reclaim the memory that the [PixelShader](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [PixelShader](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [PixelShader](#) object's [Finalize](#) method.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [PixelShader](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.GetShaderCode Method

Gets an array of bytes containing the shader code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte[] GetShaderCode ()
```

Return Value

The shader code.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the Disposing event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelShader.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[PixelShader Class](#)





[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the GraphicsDevice associated with this PixelShader .
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Name	Returns the name of the pixel shader.
	Tag	Gets the resource tags.

See Also

Reference

[PixelShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PixelShader.GraphicsDevice Property

Gets the [GraphicsDevice](#) associated with this [PixelShader](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The [GraphicsDevice](#) associated with this [PixelShader](#).

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.Name Property

Returns the name of the pixel shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of the pixel shader.

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader.Tag Property

Gets the resource tags.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tags

See Also

Reference

[PixelShader Class](#)


[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PixelShader Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[PixelShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PixelShader.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[PixelShader Class](#)

[PixelShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PresentationParameters Class

Contains presentation parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public class PresentationParameters : IDisposable
```

See Also

Concepts

[What Is Antialiasing?](#)

Tasks

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[PresentationParameters Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


PresentationParameters Members

The following tables list the members exposed by the PresentationParameters type.
















Public Constructors

Name	Description
 PresentationParameters	Initializes a new instance of this class.











Public Fields

Name	Description
 DefaultPresentRate	The default presentation rate.



Public Properties

Name	Description
 AutoDepthStencilFormat	Gets or sets a value specifying the format of the depth-stencil surface that the device will automatically create.
 BackBufferCount	Gets or sets a value specifying how many back buffers to create.
 BackBufferFormat	Gets or sets the format of the back buffer.
 BackBufferHeight	Gets or sets a value indicating the height of the new swap chain's back buffer.
 BackBufferWidth	Gets or sets a value indicating the width of the new swap chain's back buffer.
 DeviceWindowHandle	Gets or sets the handle to the device window.
 EnableAutoDepthStencil	Gets or sets a value indicating whether Direct3D will manage depth buffers for the application.
 FullScreenRefreshRateInHz	Gets or sets a value indicating the rate at which the display adapter will refresh the screen.
 IsFullScreen	Gets or sets a value indicating whether the application is in full screen mode.
 MultiSampleQuality	Gets or sets a value indicating the multisample quality level.
 MultiSampleType	Gets or sets the multisample type.
 PresentationInterval	Gets or sets the maximum rate at which the swap chain's back buffers can be presented to the front buffer.
 PresentOptions	Gets or sets miscellaneous presentation flags.
 RenderTargetUsage	Gets or sets render target usage flags.
 SwapEffect	Gets or sets the swap effect.

Public Methods

Name	Description
 Clear	Resets all of the PresentationParameters values.
 Clone	Creates a copy of this PresentationParameters object.
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Sop_Equality	Compares two objects to determine whether they are the same.
 Sop_Inequality	Compares two objects to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also


Reference

[PresentationParameters Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PresentationParameters Fields

Public Fields

	Name	Description
	DefaultPresentRate	The default presentation rate.

See Also

Reference

[PresentationParameters Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PresentationParameters.DefaultPresentRate Field

The default presentation rate.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int DefaultPresentRate
```

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PresentationParameters ()
```

See Also

Reference

[PresentationParameters Class](#)













[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters Methods

Public Methods

	Name	Description
	Clear	Resets all of the PresentationParameters values.
	Clone	Creates a copy of this PresentationParameters object.
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares two objects to determine whether they are the same.
	 op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PresentationParameters Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PresentationParameters.Clear Method

Resets all of the [PresentationParameters](#) values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear ()
```

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.Clone Method

Creates a copy of this [PresentationParameters](#) object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PresentationParameters Clone ()
```

Return Value

A copy of this [PresentationParameters](#) object.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
PresentationParameters.Dispose ()	Immediately releases the unmanaged resources used by this object.
PresentationParameters.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PresentationParameters.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [PresentationParameters](#). The [Dispose](#) method leaves the [PresentationParameters](#) in an unusable state. After calling [Dispose](#), you must release all references to the [PresentationParameters](#) so the garbage collector can reclaim the memory that the [PresentationParameters](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [PresentationParameters](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [PresentationParameters](#) object's **Finalize** method.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the **Finalize** method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. **Finalize** invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [PresentationParameters](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
PresentationParameters.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
PresentationParameters.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PresentationParameters.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [PresentationParameters](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    PresentationParameters left,  
    PresentationParameters right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    PresentationParameters left,  
    PresentationParameters right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[PresentationParameters Class](#)
















[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters Properties

Public Properties

	Name	Description
	AutoDepthStencilFormat	Gets or sets a value specifying the format of the depth-stencil surface that the device will automatically create.
	BackBufferCount	Gets or sets a value specifying how many back buffers to create.
	BackBufferFormat	Gets or sets the format of the back buffer.
	BackBufferHeight	Gets or sets a value indicating the height of the new swap chain's back buffer.
	BackBufferWidth	Gets or sets a value indicating the width of the new swap chain's back buffer.
	DeviceWindowHandle	Gets or sets the handle to the device window.
	EnableAutoDepthStencil	Gets or sets a value indicating whether Direct3D will manage depth buffers for the application.
	FullScreenRefreshRateInHz	Gets or sets a value indicating the rate at which the display adapter will refresh the screen.
	IsFullScreen	Gets or sets a value indicating whether the application is in full screen mode.
	MultiSampleQuality	Gets or sets a value indicating the multisample quality level.
	MultiSampleType	Gets or sets the multisample type.
	PresentationInterval	Gets or sets the maximum rate at which the swap chain's back buffers can be presented to the front buffer.
	PresentOptions	Gets or sets miscellaneous presentation flags.
	RenderTargetUsage	Gets or sets render target usage flags.
	SwapEffect	Gets or sets the swap effect.

See Also

Reference

[PresentationParameters Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PresentationParameters.AutoDepthStencilFormat Property

Gets or sets a value specifying the format of the depth-stencil surface that the device will automatically create.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DepthFormat AutoDepthStencilFormat { get; set; }
```

Property Value

The format of the depth-stencil surface that the device will automatically create.

Remarks

This member is ignored unless [EnableAutoDepthStencil](#) is **true**.

⚠ Caution

In cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is used on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions. Using a depth-stencil surface directly affects the size of the back buffer and depth-stencil buffer and may trigger predicated tiling.

See Also

Concepts

[What Is a Depth Buffer?](#)

[What Is a Stencil Buffer?](#)

[Predicated Tiling](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.BackBufferCount Property

Gets or sets a value specifying how many back buffers to create.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BackBufferCount { get; set; }
```

Property Value

The number of back buffers to create. Can be 0, 1, 2, or 3. 0 is considered 1.

Remarks

When an instance of this class is returned from a method (for example, the [GraphicsDevice](#) constructor), **BackBufferCount** is set to the number of back buffers that can be created. For that reason, an application can call the method twice and expect the second call to successfully create back buffers.

[SwapEffect.Copy](#) requires exactly one back buffer.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.BackBufferFormat Property

Gets or sets the format of the back buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat BackBufferFormat { get; set; }
```

Property Value

The format of the back buffer.

Remarks

For windowed applications, the back-buffer format no longer needs to match the display-mode format because color conversion can now be done by the hardware (if the hardware supports color conversion). The set of possible back-buffer formats is constrained, but the runtime will allow any valid back-buffer format to be presented to any desktop format. (There is the additional requirement that the device be operable in the desktop mode; devices typically do not operate in 8-bits-per-pixel modes.)

Full screen applications cannot do color conversion.

⚠ Caution

In cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions. The format of the back buffer chosen with the **BackBufferFormat** property directly affects the size of the back buffer and may trigger predicated tiling.

See Also

Concepts

[Predicated Tiling](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.BackBufferHeight Property

Gets or sets a value indicating the height of the new swap chain's back buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BackBufferHeight { get; set; }
```

Property Value

The height of the back buffer, in pixels.

Remarks

If [IsFullScreen](#) is **true**, [BackBufferHeight](#) and [BackBufferWidth](#) must equal the width and height of one of the enumerated display modes found through [SupportedDisplayModes](#). If [IsFullScreen](#) is **false** and either of these values is zero, the corresponding dimension of the client area of the [DeviceWindowHandle](#) (or the focus window, if [DeviceWindowHandle](#) is [Zero](#)) is taken.

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Tasks

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.BackBufferWidth Property

Gets or sets a value indicating the width of the new swap chain's back buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BackBufferWidth { get; set; }
```

Property Value

The width of the back buffer, in pixels.

Remarks

If [IsFullScreen](#) is **true**, [BackBufferHeight](#) and [BackBufferWidth](#) must equal the width and height of one of the enumerated display modes found through [SupportedDisplayModes](#). If [IsFullScreen](#) is **false** and either of these values is zero, the corresponding dimension of the client area of the [DeviceWindowHandle](#) (or the focus window, if [DeviceWindowHandle](#) is [Zero](#)) is taken.

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Tasks

[How To: Restrict Graphics Devices to Widescreen Aspect Ratios in Full-Screen Mode](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.DeviceWindowHandle Property

Gets or sets the handle to the device window.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IntPtr DeviceWindowHandle { get; set; }
```

Property Value

The handle to the device window.

Remarks

The device window determines the location and size of the back buffer on screen. This is used by Direct3D when the back buffer contents are copied to the front buffer during [Present](#).

For a full screen application, this is a handle to the top window (which is the focus window). For applications that use multiple full screen devices (such as a multimonitor system), exactly one device can use the focus window as the device window. All other devices must have unique device windows.

For a windowed-mode application, this handle will be the default target window for [Present](#). If this handle is [Zero](#), the focus window will be taken.

Note that the runtime does not attempt to reflect user changes in window size. The back buffer is not implicitly reset when this window is reset. However, the [Present](#) method does automatically track window position changes.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.EnableAutoDepthStencil Property

Gets or sets a value indicating whether Direct3D will manage depth buffers for the application.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool EnableAutoDepthStencil { get; set; }
```

Property Value

true if Direct3D will manage depth buffers for the application; **false** otherwise.

Remarks

If [EnableAutoDepthStencil](#) is **true**, the device will create a depth-stencil buffer when it is created. The depth-stencil buffer will be automatically set as the render target of the device. When the device is reset, the depth-stencil buffer will be automatically destroyed and recreated in the new size.

If [EnableAutoDepthStencil](#) is **true**, then [AutoDepthStencilFormat](#) must be a valid depth-stencil format.

See Also

Concepts

[What Is a Depth Buffer?](#)

[What Is a Stencil Buffer?](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.FullScreenRefreshRateInHz Property

Gets or sets a value indicating the rate at which the display adapter will refresh the screen.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int FullScreenRefreshRateInHz { get; set; }
```

Property Value

The refresh rate in hertz, or 0 in windowed mode.

Remarks

For full screen mode, the refresh rate must be one of the refresh rates found in [SupportedDisplayModes](#).

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.IsFullScreen Property

Gets or sets a value indicating whether the application is in full screen mode.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsFullScreen { get; set; }
```

Property Value

true if the application is running full screen; **false** if the application is running in a window.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.MultiSampleQuality Property

Gets or sets a value indicating the multisample quality level.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MultiSampleQuality { get; set; }
```

Property Value

The multisample quality level. The valid range is from 0 to one less than the value returned by [CheckDeviceMultiSampleType](#).

Remarks

Paired values of render targets or of depth-stencil surfaces and [MultiSampleType](#) must match.

Example

Use the [PreparingDeviceSettings](#) event to set **MultiSampleQuality** after calling [CheckDeviceMultiSampleType](#):

C#

```
void graphics_PreparingDeviceSettings(object sender,
    PreparingDeviceSettingsEventArgs e)
{
    // Xbox 360 and most PCs support FourSamples/0
    // (4x) and TwoSamples/0 (2x) antialiasing.
    PresentationParameters pp =
        e.GraphicsDeviceInformation.PresentationParameters;

    #if XBOX
        pp.MultiSampleQuality = 0;
        pp.MultiSampleType = MultiSampleType.FourSamples;
        return;
    #else
        int quality = 0;
        GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
        SurfaceFormat format = adapter.CurrentDisplayMode.Format;
        // Check for 4xAA
        if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
            false, MultiSampleType.FourSamples, out quality))
        {
            // even if a greater quality is returned, we only want quality 0
            pp.MultiSampleQuality = 0;
            pp.MultiSampleType =
                MultiSampleType.FourSamples;
        }
        // Check for 2xAA
        else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
            format, false, MultiSampleType.TwoSamples, out quality))
        {
            // even if a greater quality is returned, we only want quality 0
            pp.MultiSampleQuality = 0;
            pp.MultiSampleType =
                MultiSampleType.TwoSamples;
        }
        return;
    #endif
}
```

See Also

Concepts

[What Is Antialiasing?](#)

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[CheckDeviceMultiSampleType](#)

[PreparingDeviceSettings](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.MultiSampleType Property

Gets or sets the multisample type.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MultiSampleType MultiSampleType { get; set; }
```

Property Value

The multisample type. Must be [MultiSampleType.None](#) unless [SwapEffect](#) has been set to [SwapEffect.Discard](#). Multisampling is supported only if [SwapEffect](#) is [SwapEffect.Discard](#).

Remarks

⚠Caution

In cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), the platform uses [predicated tiling](#) to compensate for the additional memory requirements. Predicated tiling renders scenes performed multiple times on subsections of the final render target dimensions. The multisampling chosen with the **MultiSampleType** property directly affects the size of the back buffer and may trigger predicated tiling.

Example

Use the [PreparingDeviceSettings](#) event to set **MultiSampleType** after calling [CheckDeviceMultiSampleType](#):

C#

```
void graphics_PreparingDeviceSettings(object sender,
    PreparingDeviceSettingsEventArgs e)
{
    // Xbox 360 and most PCs support FourSamples/0
    // (4x) and TwoSamples/0 (2x) antialiasing.
    PresentationParameters pp =
        e.GraphicsDeviceInformation.PresentationParameters;
#if XBOX
    pp.MultiSampleQuality = 0;
    pp.MultiSampleType = MultiSampleType.FourSamples;
    return;
#else
    int quality = 0;
    GraphicsAdapter adapter = e.GraphicsDeviceInformation.Adapter;
    SurfaceFormat format = adapter.CurrentDisplayMode.Format;
    // Check for 4xAA
    if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware, format,
        false, MultiSampleType.FourSamples, out quality))
    {
        // even if a greater quality is returned, we only want quality 0
        pp.MultiSampleQuality = 0;
        pp.MultiSampleType =
            MultiSampleType.FourSamples;
    }
    // Check for 2xAA
    else if (adapter.CheckDeviceMultiSampleType(DeviceType.Hardware,
        format, false, MultiSampleType.TwoSamples, out quality))
    {
        // even if a greater quality is returned, we only want quality 0
        pp.MultiSampleQuality = 0;
        pp.MultiSampleType =
            MultiSampleType.TwoSamples;
    }
    return;
#endif
}
```

See Also

Concepts

[Predicated Tiling](#)

Concepts

[What Is Antialiasing?](#)

Tasks

[How To: Enable Antialiasing \(Multisampling\)](#)

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[CheckDeviceMultiSampleType](#)

[PreparingDeviceSettings](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

[Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune](#)

PresentationParameters.PresentationInterval Property

Gets or sets the maximum rate at which the swap chain's back buffers can be presented to the front buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PresentInterval PresentationInterval { get; set; }
```

Property Value

The maximum rate at which the swap chain's back buffers can be presented to the front buffer.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.PresentOptions Property

Gets or sets miscellaneous presentation flags.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PresentOptions PresentOptions { get; set; }
```

Property Value

Presentation flags.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentationParameters.RenderTargetUsage Property

Gets or sets render target usage flags.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTargetUsage RenderTargetUsage { get; set; }
```

Property Value

Usage flags for the render target.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

PresentationParameters.SwapEffect Property

Gets or sets the swap effect.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SwapEffect SwapEffect { get; set; }
```

Property Value

The swap effect to use.

Remarks

The runtime will guarantee the implied semantics concerning buffer swap behavior. Therefore, if [IsFullScreen](#) is **false** and [SwapEffect](#) is set to [SwapEffect.Flip](#), the runtime will create one extra back buffer and copy whichever becomes the front buffer at presentation time.

[SwapEffect.Copy](#) requires that [BackBufferCount](#) be set to 1.

[SwapEffect.Discard](#) will be enforced in the debug runtime by filling any buffer with noise after it is presented.

See Also

Reference

[PresentationParameters Class](#)

[PresentationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentInterval Enumeration

Defines flags that describe the relationship between the adapter refresh rate and the rate at which [Present](#) operations are completed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum PresentInterval
```

Members

Member name	Description
Default	Equivalent to setting One .
One	The driver waits for the vertical retrace period (the runtime will beam trace to prevent tearing). Present operations are not affected more frequently than the screen refresh rate; the runtime completes one Present operation per adapter refresh period, at most. This option is always available for both windowed and full-screen swap chains.
Two	The driver waits for the vertical retrace period. Present operations are not affected more frequently than every second screen refresh. Check the PresentInterval property to determine whether the driver supports this option.
Three	The driver waits for the vertical retrace period. Present operations are not affected more frequently than every third screen refresh. Check the PresentInterval property to determine whether the driver supports this option.
Four	The driver waits for the vertical retrace period. Present operations are not affected more frequently than every fourth screen refresh. Check the PresentInterval property to determine whether the driver supports this option.
Immediate	The runtime updates the window client area immediately, and might do so more than once during the adapter refresh period. Present operations might be affected immediately. This option is always available for both windowed and full-screen swap chains.

See Also

Reference

[GraphicsDeviceCapabilities.PresentInterval Property](#)

[PresentationParameters.PresentationInterval Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PresentOptions Enumeration

Defines flags that control the behavior of the back buffer and depth buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum PresentOptions
```

Members

Member name	Description
DeviceClip	Clips a windowed Present blit into the window client area, within the monitor screen area of the video adapter that created the Microsoft Direct3D device. This flag works only on Microsoft Windows 2000 and Windows XP.
DiscardDepthStencil	<p>Enables depth-buffer discarding if set when the device or swap chain is created. When this flag is set, the contents of the depth stencil buffer are invalid after either Present or DepthStencilBuffer is called.</p> <p>Discarding depth-buffer data can increase performance and is dependent on the driver. The debug runtime enforces discarding by clearing the depth-buffer to some constant value after calling either Present or DepthStencilBuffer with a different depth surface.</p> <p>Discarding depth-buffer data is illegal for all lockable formats, DepthFormat.Depth16Lockable, and DepthFormat.Depth32Lockable. Using the GraphicsDevice constructor to specify a lockable format and depth-buffer discarding will result in failure.</p>
None	Use no presentation flags.
Video	Informs the driver that the back buffers contain video data.

See Also

Reference

[PresentationParameters.PresentOptions Property](#)

[DepthFormat Enumeration](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PrimitiveType Enumeration

Defines how data in a vertex stream is interpreted during a draw call.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum PrimitiveType
```

Members

Member name	Description
LineList	Renders the vertices as a list of isolated straight line segments; the count may be any positive integer.
LineStrip	Renders the vertices as a single polyline; the count may be any positive integer.
PointList	Renders the vertices as a collection of isolated points. This value is unsupported for indexed primitives.
TriangleFan	Renders the vertices as a triangle fan.
TriangleList	Renders the specified vertices as a sequence of isolated triangles. Each group of three vertices defines a separate triangle. Back-face culling is affected by the current winding-order render state.
TriangleStrip	Renders the vertices as a triangle strip. The back-face culling flag is flipped automatically on even-numbered triangles.

Remarks Using triangle strips or triangle fans is often more efficient than using triangle lists because fewer vertices are duplicated.

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[GraphicsDevice.DrawIndexedPrimitives Method](#)

[GraphicsDevice.DrawPrimitives Method](#)

[GraphicsDevice.DrawUserIndexedPrimitives Method](#)

[GraphicsDevice.DrawUserPrimitives Generic Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

QueryUsages Enumeration

Defines options for querying device resource formats.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum QueryUsages
```

Members

Member name	Description
Filter	Query the resource format to see if it supports texture filter types other than TextureFilter.Point (which is always supported).
None	No option specified.
PostPixelShaderBlending	Query the resource to verify support for post pixel shader blending support, including alpha test, pixel fog, render-target blending, color write enable, and dithering.
SrgbRead	Query the resource to verify if a texture supports gamma correction during a read operation.
SrgbWrite	Query the resource to verify if a texture supports gamma correction during a write operation.
VertexTexture	Query the resource to verify support for vertex shader texture sampling.
WrapAndMip	Query the resource to verify support for texture wrapping and mipmapping.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[GraphicsAdapter.CheckDeviceFormat](#) Method

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RasterStatus Structure

Describes the raster status.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct RasterStatus
```

See Also

Reference

[RasterStatus Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





RasterStatus Members

The following tables list the members exposed by the RasterStatus type.



Public Properties

	Name	Description
	InVerticalBlank	Gets the status of the vertical blank period.
	ScanLine	Gets a value that roughly corresponds to the current scan line painted by the raster.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[RasterStatus Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

RasterStatus Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[RasterStatus Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RasterStatus.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[RasterStatus Structure](#)



[RasterStatus Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RasterStatus Properties

Public Properties

	Name	Description
	InVerticalBlank	Gets the status of the vertical blank period.
	ScanLine	Gets a value that roughly corresponds to the current scan line painted by the raster.

See Also

Reference

[RasterStatus Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RasterStatus.InVerticalBlank Property

Gets the status of the vertical blank period.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool InVerticalBlank { get; }
```

Property Value

true if the raster is in the vertical blank period; **false** otherwise.

See Also

Reference

[RasterStatus Structure](#)

[RasterStatus Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RasterStatus.ScanLine Property

Gets a value that roughly corresponds to the current scan line painted by the raster.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ScanLine { get; }
```

Property Value

Integer that roughly corresponds to the current scan line to set or retrieve.

Remarks

If [InVerticalBlank](#) is **false**, this property's value is an integer that roughly corresponds to the current scan line painted by the raster. Scan lines are numbered in the same way as Microsoft Direct3D surface coordinates: 0 is the top of the primary surface, extending to the value (height of the surface - 1) at the bottom of the display.

If [InVerticalBlank](#) is **true**, this value is set to 0 and can be ignored.

See Also

Reference

[RasterStatus Structure](#)

[RasterStatus Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState Class

Defines the render state of a graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class RenderState
```

RemarksThe render state of the device may also be changed by setting the render state in an effect file. For more information, please see [Effect States](#).

See Also

Concepts

[What Is Color Blending?](#)

Reference

[RenderState Members](#)



























[Microsoft.Xna.Framework.Graphics Namespace](#)
































PlatformsXbox 360, Windows XP SP2, Windows Vista







RenderState Members

The following tables list the members exposed by the RenderState type.






Public Properties

Name	Description
 AlphaBlendEnable	Gets or sets a value to enable alpha-blended transparency. The default value is false.
 AlphaBlendOperation	Gets or sets the arithmetic operation applied to separate alpha blending. The default is BlendFunction.Add .
 AlphaDestinationBlend	Gets or sets the separate alpha channel blending factor. This factor represents a destination value by which to multiply the alpha channel only. The default is Blend.One .
 AlphaFunction	Gets or sets the comparison function for the alpha test. The default is CompareFunction.Always .
 AlphaSourceBlend	Gets or sets the separate alpha channel blending factor. This factor represents a value by which to multiply the alpha channel only. The default is Blend.One .
 AlphaTestEnable	Gets or sets a render state that enables a per-pixel alpha test. The default value is false.
 BlendFactor	Gets or sets the color used for a constant-blend factor during alpha blending. The default is Color.White .
 BlendFunction	Gets or sets a value to select the arithmetic operation to apply to the source and destination pixel components when AlphaBlendEnable is set to true. The default is BlendFunction.Add .
 ColorWriteChannels	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 ColorWriteChannels1	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 ColorWriteChannels2	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 ColorWriteChannels3	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 CounterClockwiseStencilDepthBufferFail	Gets or sets the stencil operation to perform if the stencil test passes and the depth-buffer test fails for a counterclockwise triangle. The default is StencilOperation.Keep .
 CounterClockwiseStencilFail	Gets or sets the stencil operation to perform if the stencil test fails for a counterclockwise triangle. The default is StencilOperation.Keep .
 CounterClockwiseStencilFunction	Gets or sets the comparison function to use for counterclockwise stencil tests. The default is CompareFunction.Always .
 CounterClockwiseStencilPass	Gets or sets the stencil operation to perform if the stencil and z-tests pass for a counterclockwise triangle. The default is StencilOperation.Keep .
 CullMode	Specifies how back-facing triangles are culled, if at all. The default value is CullMode.CounterClockwise .
 DepthBias	Sets or retrieves the depth bias for polygons. The default value is 0.
 DepthBufferEnable	Enables or disables depth buffering. The default is true.
 DepthBufferFunction	Gets or sets the comparison function for the depth-buffer test. The default is CompareFunction.LessEqual .
 DepthBufferWriteEnable	Enables or disables writing to the depth buffer. The default is true.
 DestinationBlend	Gets or sets the color blending factor. This factor represents a value by which to multiply the destination pixel color before adding it to the source pixel to produce a color that is a blend of the two. The default is Blend.Zero .
 FillMode	Represents the fill mode. The default is FillMode.Solid .
 FogColor	Gets or sets the fog color. The default value is Color.TransparentBlack .
 FogDensity	Gets or sets the fog density for pixel or vertex fog used in exponential fog modes. The default value is 1.0f.
 FogEnable	Enables or disables fog blending. The default is false.
FogEnd	Gets or sets the depth at which pixel or vertex fog effects end for linear fog mode. The default value is 1.0f.
FogStart	Gets or sets the depth at which pixel or vertex fog effects begin for linear fog mode. The default value is 0.0f.
FogTableMode	Gets or sets the fog formula to use for pixel fog. The default is None .
FogVertexMode	Gets or sets the fog formula to use for vertex fog. The default is FogMode.None .



 MultiSampleAntiAlias	Enables or disables multisample antialiasing. The default is true.
 MultiSampleMask	Gets or sets a bitmask controlling modification of the samples in a multisample render target. The default is 0xffffffff.
 PointSize	Gets or sets the size to use for point size computation in cases where point size is not specified for each vertex. The default value is the value a driver returns. If a driver returns 0 or 1, the default value is 64, which allows software point size emulation.
 PointSizeMax	Gets or sets the maximum size of point primitives. The default is 64.0f.
 PointSizeMin	Gets or sets the minimum size of point primitives. The default is 1.0f.
 PointSpriteEnable	Enables or disables full texture mapping on each point. The default is false.
 RangeFogEnable	Gets or sets enabling of range-based vertex fog. The default value is false.
 ReferenceAlpha	Specifies a reference alpha value against which pixels are tested when alpha testing is enabled. The default value is 0.
 ReferenceStencil	Specifies a reference value to use for the stencil test. The default is 0.
 ScissorTestEnable	Enables or disables scissor testing. The default is false.
 SeparateAlphaBlendEnabled	Enables or disables the separate blend mode for the alpha channel. The default is false.
 SlopeScaleDepthBias	Gets or sets a value used to determine how much bias can be applied to coplanar primitives to reduce flinmering z-fighting. The default is 0.
 SourceBlend	Gets or sets the color blending factor. This factor represents a value by which to multiply the source pixel color before adding it to the destination pixel to produce a color that is a blend of the two. The default is Blend.One .
 StencilDepthBufferFail	Gets or sets the stencil operation to perform if the stencil test passes and the depth-test fails. The default is StencilOperation.Keep .
 StencilEnable	Gets or sets stencil enabling. The default is false.
 StencilFail	Gets or sets the stencil operation to perform if the stencil test fails. The default is StencilOperation.Keep .
 StencilFunction	Gets or sets the comparison function for the stencil test. The default is CompareFunction.Always .
 StencilMask	Gets or sets the mask applied to the reference value and each stencil buffer entry to determine the significant bits for the stencil test. The default mask is Int32.MaxValue.
 StencilPass	Gets or sets the stencil operation to perform if the stencil test passes. The default is StencilOperation.Keep .
 StencilWriteMask	Gets or sets the write mask applied to values written into the stencil buffer. The default mask is Int32.MaxValue.
 TwoSidedStencilMode	Enables or disables two-sided stenciling. The default is false.
 Wrap0	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap1	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap10	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap11	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap12	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap13	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap14	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap15	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap2	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap3	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).

 Wrap4	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap5	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap6	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap7	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap8	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap9	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[RenderState Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderState Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[RenderState Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderState.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[RenderState Class](#)




























[RenderState Members](#)
































[Microsoft.Xna.Framework.Graphics Namespace](#)






Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState Properties

Public Properties

Name	Description
 AlphaBlendEnable	Gets or sets a value to enable alpha-blended transparency. The default value is false.
 AlphaBlendOperation	Gets or sets the arithmetic operation applied to separate alpha blending. The default is BlendFunction.Add .
 AlphaDestinationBlend	Gets or sets the separate alpha channel blending factor. This factor represents a destination value by which to multiply the alpha channel only. The default is Blend.One .
 AlphaFunction	Gets or sets the comparison function for the alpha test. The default is CompareFunction.Always .
 AlphaSourceBlend	Gets or sets the separate alpha channel blending factor. This factor represents a value by which to multiply the alpha channel only. The default is Blend.One .
 AlphaTestEnable	Gets or sets a render state that enables a per-pixel alpha test. The default value is false.
 BlendFactor	Gets or sets the color used for a constant-blend factor during alpha blending. The default is Color.White .
 BlendFunction	Gets or sets a value to select the arithmetic operation to apply to the source and destination pixel components when AlphaBlendEnable is set to true. The default is BlendFunction.Add .
 ColorWriteChannels	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 ColorWriteChannels1	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 ColorWriteChannels2	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 ColorWriteChannels3	Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is ColorWriteChannels.None .
 CounterClockwiseStencilDepthBufferFail	Gets or sets the stencil operation to perform if the stencil test passes and the depth-buffer test fails for a counterclockwise triangle. The default is StencilOperation.Keep .
 CounterClockwiseStencilFail	Gets or sets the stencil operation to perform if the stencil test fails for a counterclockwise triangle. The default is StencilOperation.Keep .
 CounterClockwiseStencilFunction	Gets or sets the comparison function to use for counterclockwise stencil tests. The default is CompareFunction.Always .
 CounterClockwiseStencilPass	Gets or sets the stencil operation to perform if the stencil and z-tests pass for a counterclockwise triangle. The default is StencilOperation.Keep .
 CullMode	Specifies how back-facing triangles are culled, if at all. The default value is CullMode.CounterClockwise .
 DepthBias	Sets or retrieves the depth bias for polygons. The default value is 0.
 DepthBufferEnable	Enables or disables depth buffering. The default is true.
 DepthBufferFunction	Gets or sets the comparison function for the depth-buffer test. The default is CompareFunction.LessEqual .
 DepthBufferWriteEnable	Enables or disables writing to the depth buffer. The default is true.
 DestinationBlend	Gets or sets the color blending factor. This factor represents a value by which to multiply the destination pixel color before adding it to the source pixel to produce a color that is a blend of the two. The default is Blend.Zero .
 FillMode	Represents the fill mode. The default is FillMode.Solid .
 FogColor	Gets or sets the fog color. The default value is Color.TransparentBlack .
 FogDensity	Gets or sets the fog density for pixel or vertex fog used in exponential fog modes. The default value is 1.0f.
 FogEnable	Enables or disables fog blending. The default is false.
 FogEnd	Gets or sets the depth at which pixel or vertex fog effects end for linear fog mode. The default value is 1.0f.
FogStart	Gets or sets the depth at which pixel or vertex fog effects begin for linear fog mode. The default value is 0.0f.
FogTableMode	Gets or sets the fog formula to use for pixel fog. The default is None .
FogVertexMode	Gets or sets the fog formula to use for vertex fog. The default is FogMode.None .
MultiSampleAntiAlias	Enables or disables multisample antialiasing. The default is true.

 MultiSampleMask	Gets or sets a bitmask controlling modification of the samples in a multisample render target. The default is <code>0xffffffff</code> .
 PointSize	Gets or sets the size to use for point size computation in cases where point size is not specified for each vertex. The default value is the value a driver returns. If a driver returns 0 or 1, the default value is 64, which allows software point size emulation.
 PointSizeMax	Gets or sets the maximum size of point primitives. The default is 64.0f.
 PointSizeMin	Gets or sets the minimum size of point primitives. The default is 1.0f.
 PointSpriteEnable	Enables or disables full texture mapping on each point. The default is false.
 RangeFogEnable	Gets or sets enabling of range-based vertex fog. The default value is false.
 ReferenceAlpha	Specifies a reference alpha value against which pixels are tested when alpha testing is enabled. The default value is 0.
 ReferenceStencil	Specifies a reference value to use for the stencil test. The default is 0.
 ScissorTestEnable	Enables or disables scissor testing. The default is false.
 SeparateAlphaBlendEnabled	Enables or disables the separate blend mode for the alpha channel. The default is false.
 SlopeScaleDepthBias	Gets or sets a value used to determine how much bias can be applied to coplanar primitives to reduce flickering z-fighting. The default is 0.
 SourceBlend	Gets or sets the color blending factor. This factor represents a value by which to multiply the source pixel color before adding it to the destination pixel to produce a color that is a blend of the two. The default is <code>Blend.One</code> .
 StencilDepthBufferFail	Gets or sets the stencil operation to perform if the stencil test passes and the depth-test fails. The default is <code>StencilOperation.Keep</code> .
 StencilEnable	Gets or sets stencil enabling. The default is false.
 StencilFail	Gets or sets the stencil operation to perform if the stencil test fails. The default is <code>StencilOperation.Keep</code> .
 StencilFunction	Gets or sets the comparison function for the stencil test. The default is <code>CompareFunction.Always</code> .
 StencilMask	Gets or sets the mask applied to the reference value and each stencil buffer entry to determine the significant bits for the stencil test. The default mask is <code>Int32.MaxValue</code> .
 StencilPass	Gets or sets the stencil operation to perform if the stencil test passes. The default is <code>StencilOperation.Keep</code> .
 StencilWriteMask	Gets or sets the write mask applied to values written into the stencil buffer. The default mask is <code>Int32.MaxValue</code> .
 TwoSidedStencilMode	Enables or disables two-sided stenciling. The default is false.
 Wrap0	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap1	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap10	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap11	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap12	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap13	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap14	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap15	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap2	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap3	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).
 Wrap4	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is <code>TextureWrapCoordinates.Zero</code> (wrapping disabled in all directions).

 Wrap5	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap6	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap7	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap8	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).
 Wrap9	Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is TextureWrapCoordinates.Zero (wrapping disabled in all directions).

See Also

Reference

[RenderState Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderState.AlphaBlendEnable Property

Gets or sets a value to enable alpha-blended transparency. The default value is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AlphaBlendEnable { get; set; }
```

Property Value

true if alpha-blended transparency is enabled; **false** otherwise.

Remarks

The alpha value of a color controls its transparency. Enabling alpha blending with [AlphaBlendEnable](#) allows colors, materials, and textures on a surface to be blended with transparency onto another surface.

Alpha blending is determined by the [SourceBlend](#) and [DestinationBlend](#) properties. With the default [BlendFunction](#), the source pixel times the [SourceBlend](#) is added to the destination pixel (the pixel on the back buffer) times the [DestinationBlend](#) to produce a color that is a blend of the two pixel colors. The [BlendFunction](#) determines the operation used on the outcome of the source blend and destination blend components.

Example

The following short code snippet sets the render state to use additive alpha blending prior to a draw call.

C#

```
graphics.GraphicsDevice.RenderState.AlphaBlendEnable = true;
graphics.GraphicsDevice.RenderState.SourceBlend = Blend.One;
graphics.GraphicsDevice.RenderState.DestinationBlend = Blend.One;
graphics.GraphicsDevice.RenderState.BlendFunction = BlendFunction.Add;

// TODO: Place the drawing calls that should be alpha blended here.
```

See Also

Reference

[SourceBlend](#)

[DestinationBlend](#)

[BlendFunction](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.AlphaBlendOperation Property

Gets or sets the arithmetic operation applied to separate alpha blending. The default is [BlendFunction.Add](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BlendFunction AlphaBlendOperation { get; set; }
```

Property Value

A value from the [BlendFunction](#) enumeration.

Remarks

This value is ignored unless [SeparateAlphaBlendEnabled](#) is **true**.

⚠ Caution

This value is used for blending the alpha channel only, not for alpha blending of the entire pixel. For setting the blending mode for alpha blending operation, see [BlendFunction](#).

See Also

Reference

[RenderState.BlendFunction Property](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.AlphaDestinationBlend Property

Gets or sets the separate alpha channel blending factor. This factor represents a destination value by which to multiply the alpha channel only. The default is [Blend.One](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Blend AlphaDestinationBlend { get; set; }
```

Property Value

A value from the [Blend](#) enumeration. **BothSourceAlpha** and **BothInverseSourceAlpha** are not supported on Xbox 360.

Remarks

This value is ignored unless [SeparateAlphaBlendEnabled](#) is **true**.

⚠ Caution

This value is used for blending the alpha channel only, not for alpha blending of the entire pixel. For setting the blending mode for alpha blending of the destination color, see [DestinationBlend](#).

See Also

Concepts

[What Is Color Blending?](#)

Reference

[RenderState.DestinationBlend Property](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.AlphaFunction Property

Gets or sets the comparison function for the alpha test. The default is [CompareFunction.Always](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompareFunction AlphaFunction { get; set; }
```

Property Value

A member of the [CompareFunction](#) enumeration that represents the comparison function to set or get.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[AlphaTestEnable](#)

[ReferenceAlpha](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.AlphaSourceBlend Property

Gets or sets the separate alpha channel blending factor. This factor represents a value by which to multiply the alpha channel only. The default is [Blend.One](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Blend AlphaSourceBlend { get; set; }
```

Property Value

A value from the [Blend](#) enumeration.

Remarks

⚠Caution

This value is used for blending the alpha channel only, not for alpha blending of the entire pixel. For setting the blending mode for alpha blending of the source color, see [SourceBlend](#).

This value is ignored unless [SeparateAlphaBlendEnabled](#) is **true**.

See Also

Concepts

[What Is Color Blending?](#)

Reference

[RenderState.SourceBlend Property](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.AlphaTestEnable Property

Gets or sets a render state that enables a per-pixel alpha test. The default value is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AlphaTestEnable { get; set; }
```

Property Value

true if per-pixel alpha blending is enabled; **false** otherwise.

Remarks

If the test passes, the pixel is processed by the frame buffer. Otherwise, all frame-buffer processing is skipped for the pixel. The test is done by comparing the incoming alpha value with the reference alpha value, using the comparison function provided by [AlphaFunction](#). The reference alpha value is determined by the value set for [ReferenceAlpha](#).

See Also

Reference

[RenderState Class](#)

[AlphaFunction](#)

[ReferenceAlpha](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.BlendFactor Property

Gets or sets the color used for a constant-blend factor during alpha blending. The default is [Color.White](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color BlendFactor { get; set; }
```

Property Value

The color used for a constant-blend factor during alpha blending.

Remarks

Blend factoring during alpha blending is available if [SupportsBlendFactor](#) or [SupportsBlendFactor](#) is **true**.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.BlendFunction Property

Gets or sets a value to select the arithmetic operation to apply to the source and destination pixel components when [AlphaBlendEnable](#) is set to **true**. The default is [BlendFunction.Add](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BlendFunction BlendFunction { get; set; }
```

Property Value

The blending operation to set or get.

Remarks

The alpha value of a color controls its transparency. Enabling alpha blending with [AlphaBlendEnable](#) allows colors, materials, and textures on a surface to be blended with transparency onto another surface.

Alpha blending is determined by the [SourceBlend](#) and [DestinationBlend](#) properties. With the default [BlendFunction](#), the source pixel times the [SourceBlend](#) is added to the destination pixel (the pixel on the back buffer) times the [DestinationBlend](#) to produce a color that is a blend of the two pixel colors. The [BlendFunction](#) determines the operation used on the outcome of the source blend and destination blend components.

Example

The following short code snippet sets the render state to use additive alpha blending prior to a draw call.

C#

```
graphics.GraphicsDevice.RenderState.AlphaBlendEnable = true;
graphics.GraphicsDevice.RenderState.SourceBlend = Blend.One;
graphics.GraphicsDevice.RenderState.DestinationBlend = Blend.One;
graphics.GraphicsDevice.RenderState.BlendFunction = BlendFunction.Add;

// TODO: Place the drawing calls that should be alpha blended here.
```

See Also

Reference

[AlphaBlendEnable](#)

[SourceBlend](#)

[DestinationBlend](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.ColorWriteChannels Property

Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is [ColorWriteChannels.None](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ColorWriteChannels ColorWriteChannels { get; set; }
```

Property Value

Value of the [ColorWriteChannels](#) enumeration that specifies the color channel to set or get.

Remarks

Color writing is available if [SupportsColorWrite](#) is **true**.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.ColorWriteChannels1 Property

Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is [ColorWriteChannels.None](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ColorWriteChannels ColorWriteChannels1 { get; set; }
```

Property Value

Value of the [ColorWriteChannels](#) enumeration that specifies the color channel to set or get.

Remarks

This property works in addition to the [ColorWriteChannels](#) property. Color writing is available if [SupportsColorWrite](#) is **true**.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.ColorWriteChannels2 Property

Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is [ColorWriteChannels.None](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ColorWriteChannels ColorWriteChannels2 { get; set; }
```

Property Value

Value of the [ColorWriteChannels](#) enumeration that specifies the color channel to set or get.

Remarks

This property works in addition to the [ColorWriteChannels](#) property. Color writing is available if [SupportsColorWrite](#) is **true**.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.ColorWriteChannels3 Property

Gets or sets a value that enables a per-channel write for the render target color buffer. The default value is [ColorWriteChannels.None](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ColorWriteChannels ColorWriteChannels3 { get; set; }
```

Property Value

Value of the [ColorWriteChannels](#) enumeration that specifies the color channel to set or get.

Remarks

This property works in addition to the [ColorWriteChannels](#) property. Color writing is available if [SupportsColorWrite](#) is **true**.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.CounterClockwiseStencilDepthBufferFail Property

Gets or sets the stencil operation to perform if the stencil test passes and the depth-buffer test fails for a counterclockwise triangle. The default is [StencilOperation.Keep](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilOperation CounterClockwiseStencilDepthBufferFail { get; set; }
```

Property Value

The stencil operation to perform.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[CounterClockwiseStencilFunction](#)

[CounterClockwiseStencilPass](#)

[CounterClockwiseStencilFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.CounterClockwiseStencilFail Property

Gets or sets the stencil operation to perform if the stencil test fails for a counterclockwise triangle. The default is [StencilOperation.Keep](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilOperation CounterClockwiseStencilFail { get; set; }
```

Property Value

The stencil operation to perform.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[CounterClockwiseStencilFunction](#)

[CounterClockwiseStencilPass](#)

[CounterClockwiseStencilDepthBufferFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.CounterClockwiseStencilFunction Property

Gets or sets the comparison function to use for counterclockwise stencil tests. The default is [CompareFunction.Always](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompareFunction CounterClockwiseStencilFunction { get; set; }
```

Property Value

A [CompareFunction](#) value indicating which test to perform.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[CounterClockwiseStencilPass](#)

[CounterClockwiseStencilFail](#)

[CounterClockwiseStencilDepthBufferFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.CounterClockwiseStencilPass Property

Gets or sets the stencil operation to perform if the stencil and z-tests pass for a counterclockwise triangle. The default is [StencilOperation.Keep](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilOperation CounterClockwiseStencilPass { get; set; }
```

Property Value

The stencil operation to perform.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[CounterClockwiseStencilFunction](#)

[CounterClockwiseStencilFail](#)

[CounterClockwiseStencilDepthBufferFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.CullMode Property

Specifies how back-facing triangles are culled, if at all. The default value is [CullMode.CounterClockwise](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CullMode CullMode { get; set; }
```

Property Value

The culling mode to set or get.

Remarks

When drawing sprites, [SpriteBatch.Begin](#) does not save your current render state, and will change certain render state properties that may make 3D objects render incorrectly. This includes setting **CullMode** to [CullMode.CullCounterClockwiseFace](#). You can choose to either reset the render state yourself after the call to [SpriteBatch.End](#), or call [SpriteBatch.Begin](#) and pass in [SaveStateMode.SaveState](#), which will restore the render state after sprites are drawn.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.DepthBias Property

Sets or retrieves the depth bias for polygons. The default value is 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float DepthBias { get; set; }
```

Property Value

Depth bias for polygons.

Remarks

This property specifies a value in the range of 0 through 16 (inclusive). This value is used to make polygons that are physically coplanar appear separate.

Polygons with a high z-bias value appear in front of polygons with a lower value, without requiring sorting for drawing order. For example, polygons with a value of 1 appear in front of polygons with a value of 0.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[SlopeScaleDepthBias](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.DepthBufferEnable Property

Enables or disables depth buffering. The default is **true**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool DepthBufferEnable { get; set; }
```

Property Value

true if depth buffering is enabled; **false** otherwise.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.DepthBufferFunction Property

Gets or sets the comparison function for the depth-buffer test. The default is [CompareFunction.LessEqual](#)

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompareFunction DepthBufferFunction { get; set; }
```

Property Value

Value of a [CompareFunction](#) that represents the comparison function to set or get.

Remarks

The depth value of the pixel is compared to the depth-buffer value. If the depth value of the pixel passes the comparison function, the pixel is written.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.DepthBufferWriteEnable Property

Enables or disables writing to the depth buffer. The default is **true**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool DepthBufferWriteEnable { get; set; }
```

Property Value

true if writing to the depth buffer is enabled; **false** otherwise.

Remarks

This property enables an application to prevent the system from updating the depth buffer with new depth values.

If **false**, depth comparisons are still made according to the render state [DepthBufferFunction](#), assuming that depth buffering is taking place, but depth values are not written to the buffer.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.DestinationBlend Property

Gets or sets the color blending factor. This factor represents a value by which to multiply the destination pixel color before adding it to the source pixel to produce a color that is a blend of the two. The default is [Blend.Zero](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Blend DestinationBlend { get; set; }
```

Property Value

A [Blend](#) factor to set or get for the destination pixel.

Remarks

The alpha value of a color controls its transparency. Enabling alpha blending with [AlphaBlendEnable](#) allows colors, materials, and textures on a surface to be blended with transparency onto another surface.

Alpha blending is determined by the [SourceBlend](#) and [DestinationBlend](#) properties. With the default [BlendFunction](#), the source pixel times the [SourceBlend](#) is added to the destination pixel (the pixel on the back buffer) times the [DestinationBlend](#) to produce a color that is a blend of the two pixel colors. The [BlendFunction](#) determines the operation used on the outcome of the source blend and destination blend components.

Example

The following short code snippet sets the render state to use additive alpha blending prior to a draw call.

C#

```
graphics.GraphicsDevice.RenderState.AlphaBlendEnable = true;  
graphics.GraphicsDevice.RenderState.SourceBlend = Blend.One;  
graphics.GraphicsDevice.RenderState.DestinationBlend = Blend.One;  
graphics.GraphicsDevice.RenderState.BlendFunction = BlendFunction.Add;  
  
// TODO: Place the drawing calls that should be alpha blended here.
```

See Also

Concepts

[What Is Color Blending?](#)

Reference

[AlphaBlendEnable](#)

[SourceBlend](#)

[BlendFunction](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.FillMode Property

Represents the fill mode. The default is [FillMode.Solid](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FillMode FillMode { get; set; }
```

Property Value

Value of a [FillMode](#) that specifies the fill mode to set or get.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.FogColor Property

Note

This property is available only when developing for Windows.

Gets or sets the fog color. The default value is [Color.TransparentBlack](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color FogColor { get; set; }
```

Property Value

A color that specifies the fog color to set or get.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogStart](#)

[FogEnd](#)

[FogEnable](#)

[FogDensity](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.FogDensity Property

Note

This property is available only when developing for Windows.

Gets or sets the fog density for pixel or vertex fog used in exponential fog modes. The default value is 1.0f.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float FogDensity { get; set; }
```

Property Value

Value that represents the fog density to set or get.

Remarks

Valid fog density values range from 0.0 through 1.0.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogStart](#)

[FogEnd](#)

[FogEnable](#)

[FogColor](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.FogEnable Property

Note

This property is available only when developing for Windows.

Enables or disables fog blending. The default is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool FogEnable { get; set; }
```

Property Value

true if fog blending is enabled; **false** otherwise.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogColor](#)

[FogVertexMode](#)

[FogStart](#)

[FogEnd](#)

[FogDensity](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.FogEnd Property

Note

This property is available only when developing for Windows.

Gets or sets the depth at which pixel or vertex fog effects end for linear fog mode. The default value is 1.0f.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float FogEnd { get; set; }
```

Property Value

Value that represents the ending depth to set or get.

Remarks

Depth is specified in world space for vertex fog, and in either device space [0.0, 1.0] or world space for pixel fog. For pixel fog, these values are in device space when the system uses z for fog calculations, or in world space when the system uses eye-relative fog (w-fog).

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogStart](#)

[FogEnable](#)

[FogColor](#)

[FogDensity](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.FogStart Property

Note

This property is available only when developing for Windows.

Gets or sets the depth at which pixel or vertex fog effects begin for linear fog mode. The default value is 0.0f.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float FogStart { get; set; }
```

Property Value

Value that represents the beginning depth to set or get.

Remarks

Depth is specified in world space for vertex fog, and in either device space [0.0, 1.0] or world space for pixel fog. For pixel fog, these values are in device space when the system uses z for fog calculations, or in world space when the system uses eye-relative fog (w-fog).

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogEnd](#)

[FogEnable](#)

[FogColor](#)

[FogDensity](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.FogTableMode Property

Note

This property is available only when developing for Windows.

Gets or sets the fog formula to use for pixel fog. The default is [None](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FogMode FogTableMode { get; set; }
```

Property Value

Value of a [FogMode](#) that specifies the fog mode to set or get.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogStart](#)

[FogEnd](#)

[FogEnable](#)

[FogColor](#)

[FogDensity](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.FogVertexMode Property

Note

This property is available only when developing for Windows.

Gets or sets the fog formula to use for vertex fog. The default is [FogMode.None](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FogMode FogVertexMode { get; set; }
```

Property Value

Value of a [FogMode](#) that specifies the fog mode to set or get.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogStart](#)

[FogEnd](#)

[FogEnable](#)

[FogColor](#)

[FogDensity](#)

[RangeFogEnable](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.MultiSampleAntiAlias Property

Enables or disables multisample antialiasing. The default is **true**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MultiSampleAntiAlias { get; set; }
```

Property Value

true to enable multisample antialiasing; **false** otherwise.

Remarks

When multisample antialiasing is enabled, the multiple samples are computed so that full-scene antialiasing is performed by sampling at different sample positions for each multiple sample. When disabled, the multiple samples are all written with the same sample value, sampled at the pixel center, which allows non-antialiased rendering to a multisample buffer. This render state has no effect when rendering to a single sample buffer.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.MultiSampleMask Property

Gets or sets a bitmask controlling modification of the samples in a multisample render target. The default is 0xffffffff.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MultiSampleMask { get; set; }
```

Property Value

A bitmask value controlling write enables for the samples. Each bit in this mask, starting at the least-significant bit, controls modification of one of the samples in a multisample render target. Thus, for an 8-sample render target, the low byte contains the eight write enables for each of the eight samples. This render state has no effect when rendering to a single sample buffer.

Remarks

This render state enables use of a multisample buffer as an accumulation buffer, doing multipass rendering of geometry where each pass updates a subset of samples.

Given n multisamples and k enabled samples, the resulting intensity of the rendered image should be k/n . Each component RGB of every pixel is factored by k/n .

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.PointSize Property

Gets or sets the size to use for point size computation in cases where point size is not specified for each vertex. The default value is the value a driver returns. If a driver returns 0 or 1, the default value is 64, which allows software point size emulation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float PointSize { get; set; }
```

Property Value

This value is in world space units.

See Also

Tasks

[How To: Draw Point Sprites](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[PointSpriteEnable](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.PointSizeMax Property

Gets or sets the maximum size of point primitives. The default is 64.0f.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float PointSizeMax { get; set; }
```

Property Value

The maximum size of point primitives. Must be less than or equal to [Capabilities.MaxPointSize](#) and greater than or equal to [PointSizeMin](#).

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.PointSizeMin Property

Gets or sets the minimum size of point primitives. The default is 1.0f.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float PointSizeMin { get; set; }
```

Property Value

The minimum size of point primitives.

Remarks

Point primitives are clamped to this size during rendering. Setting this to values smaller than 1.0 results in points dropping out when the point does not cover a pixel center and antialiasing is disabled or, when antialiasing is enabled, being rendered with reduced intensity.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.PointSpriteEnable Property

Enables or disables full texture mapping on each point. The default is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool PointSpriteEnable { get; set; }
```

Property Value

true to set texture coordinates of point primitives so that full textures are mapped on each point; **false** otherwise. When **false**, the vertex texture coordinates are used for the entire point.

See Also

Tasks

[How To: Draw Point Sprites](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[PointSize](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.RangeFogEnable Property

Note

This property is available only when developing for Windows.

Gets or sets enabling of range-based vertex fog. The default value is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool RangeFogEnable { get; set; }
```

Property Value

true if range-based vertex fog is enabled; **false** otherwise. If **false**, depth-based fog is used.

Remarks

In range-based fog, the distance of an object from the viewer is used to compute fog effects, not the depth of the object in the scene. In range-based fog, all fog methods work as usual, except that they use range instead of depth in the computations.

Range is the correct factor to use for fog computations, but depth is commonly used instead because range is time-consuming to compute and depth is generally already available. Using depth to calculate fog has the undesirable effect of having the fogginess of peripheral objects change as the viewer's eye moves - in this case, the depth changes and the range remains constant.

Because no hardware currently supports per-pixel range-based fog, range correction is offered only for vertex fog.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[FogStart](#)

[FogEnd](#)

[FogEnable](#)

[FogColor](#)

[FogDensity](#)

[FogVertexMode](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderState.ReferenceAlpha Property

Specifies a reference alpha value against which pixels are tested when alpha testing is enabled. The default value is 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ReferenceAlpha { get; set; }
```

Property Value

Integer that specifies the reference alpha value to set or get. This is an 8-bit value placed in the low 8 bits of the DWORD render-state value. Values can range from 0x00000000 through 0x000000FF.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[AlphaTestEnable](#)

[AlphaFunction](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.ReferenceStencil Property

Specifies a reference value to use for the stencil test. The default is 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ReferenceStencil { get; set; }
```

Property Value

Integer that specifies the stencil test value to set or get.

Remarks

The reference value is compared, by the comparison function specified by the [StencilFunction](#) property, to the stencil buffer entry of a pixel. This can be illustrated by a simple equation:

$$\text{ReferenceStencil} \text{ StencilFunction} (\textit{stencil buffer entry})$$

This comparison applies only to the bits in the reference value and stencil buffer entry that are set in the stencil mask by this property. If the comparison is true, the stencil test passes and the pass operation (specified by the [StencilPass](#) property) is performed.

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilEnable](#)

[StencilFunction](#)

[StencilMask](#)

[StencilWriteMask](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.ScissorTestEnable Property

Enables or disables scissor testing. The default is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool ScissorTestEnable { get; set; }
```

Property Value

true to enable scissor testing; **false** otherwise.

Remarks

With scissor testing enabled, the graphics device will only draw onto parts of the screen that fall within the specified [ScissorRectangle](#).

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[ScissorRectangle](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.SeparateAlphaBlendEnabled Property

Enables or disables the separate blend mode for the alpha channel. The default is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool SeparateAlphaBlendEnabled { get; set; }
```

Property Value

true to enable the separate blend mode for the alpha channel; **false** otherwise.

Remarks

When set to **false**, the render-target blending factors and operations applied to alpha are forced to be the same as those defined for color. This mode is effectively hard-wired to **false** on implementations that don't set [SupportsSeparateAlphaBlend](#) to **true**.

The [AlphaSourceBlend](#) and [AlphaDestinationBlend](#) render states determine the type of separate alpha blending.

See Also

Reference

[AlphaSourceBlend](#)

[AlphaDestinationBlend](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.SlopeScaleDepthBias Property

Gets or sets a value used to determine how much bias can be applied to coplanar primitives to reduce flimmering z-fighting. The default is 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float SlopeScaleDepthBias { get; set; }
```

Property Value

Value that specifies the slope scale bias to apply.

Remarks

Polygons that are coplanar in your 3D space can be made to appear as if they are not coplanar by adding a z-bias to each one. An application can help ensure that coplanar polygons are rendered properly by adding a bias to the z-values that the system uses when rendering sets of coplanar polygons.

The following formula shows how to calculate the bias to be applied to coplanar primitives.

$$\textit{bias} = (m \times \textbf{SlopeScaleDepthBias}) + \textbf{DepthBias}$$

Where m is the maximum depth slope of the triangle being rendered, defined as:

$$m = \textbf{max}(\textbf{abs}(\textit{delta z} / \textit{delta x}), \textbf{abs}(\textit{delta z} / \textit{delta y}))$$

See Also

Reference

[RenderState Class](#)

[RenderState Members](#)

[DepthBias](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.SourceBlend Property

Gets or sets the color blending factor. This factor represents a value by which to multiply the source pixel color before adding it to the destination pixel to produce a color that is a blend of the two. The default is [Blend.One](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Blend SourceBlend { get; set; }
```

Property Value

A [Blend](#) factor to set or get for the source pixel.

Remarks

The alpha value of a color controls its transparency. Enabling alpha blending with [AlphaBlendEnable](#) allows colors, materials, and textures on a surface to be blended with transparency onto another surface.

Alpha blending is determined by the [SourceBlend](#) and [DestinationBlend](#) properties. With the default [BlendFunction](#), the source pixel times the [SourceBlend](#) is added to the destination pixel (the pixel on the back buffer) times the [DestinationBlend](#) to produce a color that is a blend of the two pixel colors. The [BlendFunction](#) determines the operation used on the outcome of the source blend and destination blend components.

Example

The following short code snippet sets the render state to use additive alpha blending prior to a draw call.

C#

```
graphics.GraphicsDevice.RenderState.AlphaBlendEnable = true;
graphics.GraphicsDevice.RenderState.SourceBlend = Blend.One;
graphics.GraphicsDevice.RenderState.DestinationBlend = Blend.One;
graphics.GraphicsDevice.RenderState.BlendFunction = BlendFunction.Add;

// TODO: Place the drawing calls that should be alpha blended here.
```

See Also

Concepts

[What Is Color Blending?](#)

Reference

[AlphaBlendEnable](#)

[DestinationBlend](#)

[BlendFunction](#)

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilDepthBufferFail Property

Gets or sets the stencil operation to perform if the stencil test passes and the depth-test fails. The default is [StencilOperation.Keep](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilOperation StencilDepthBufferFail { get; set; }
```

Property Value

The stencil operation to perform.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilEnable Property

Gets or sets stencil enabling. The default is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool StencilEnable { get; set; }
```

Property Value

true if stenciling is enabled; **false** otherwise.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilFunction](#)

[StencilMask](#)

[StencilWriteMask](#)

[ReferenceStencil](#)

[StencilPass](#)

[StencilFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilFail Property

Gets or sets the stencil operation to perform if the stencil test fails. The default is [StencilOperation.Keep](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilOperation StencilFail { get; set; }
```

Property Value

The stencil operation to perform.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilEnable](#)

[StencilFunction](#)

[StencilMask](#)

[StencilWriteMask](#)

[ReferenceStencil](#)

[StencilPass](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilFunction Property

Gets or sets the comparison function for the stencil test. The default is [CompareFunction.Always](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public CompareFunction StencilFunction { get; set; }
```

Property Value

Value of a [CompareFunction](#) that represents the comparison function to set or get.

Remarks

The comparison function is used to compare the reference value (stored in the [ReferenceStencil](#) property) to the stencil buffer entry of a pixel. This can be illustrated by a simple equation:

```
ReferenceStencil StencilFunction (stencil buffer entry)
```

This comparison applies only to the bits in the reference value and stencil buffer entry that are set in the stencil mask by this property. If the comparison is true, the stencil test passes and the pass operation (specified by the [StencilPass](#) property) is performed.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilEnable](#)

[StencilMask](#)

[StencilWriteMask](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilMask Property

Gets or sets the mask applied to the reference value and each stencil buffer entry to determine the significant bits for the stencil test. The default mask is [Int32.MaxValue](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StencilMask { get; set; }
```

Property Value

Value that represents the mask to set or get.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilEnable](#)

[StencilFunction](#)

[StencilWriteMask](#)

[ReferenceStencil](#)

[StencilPass](#)

[StencilFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilPass Property

Gets or sets the stencil operation to perform if the stencil test passes. The default is [StencilOperation.Keep](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StencilOperation StencilPass { get; set; }
```

Property Value

The stencil operation to perform.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilEnable](#)

[StencilFunction](#)

[StencilMask](#)

[StencilWriteMask](#)

[ReferenceStencil](#)

[StencilFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.StencilWriteMask Property

Gets or sets the write mask applied to values written into the stencil buffer. The default mask is [Int32.MaxValue](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StencilWriteMask { get; set; }
```

Property Value

Value that represents the write mask to set or get.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[StencilEnable](#)

[StencilFunction](#)

[StencilMask](#)

[ReferenceStencil](#)

[StencilPass](#)

[StencilFail](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.TwoSidedStencilMode Property

Enables or disables two-sided stenciling. The default is **false**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool TwoSidedStencilMode { get; set; }
```

Property Value

true to enable two-sided stenciling; **false** otherwise.

Remarks

The application should set [CullMode](#) to [CullMode.None](#) to enable two-sided stencil mode. If the triangle winding order is clockwise, the [StencilFunction](#) will be used. If the winding order is counterclockwise, the [CounterClockwiseStencilFunction](#) will be used.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap0 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap0 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap1 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap1 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap10 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap10 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap11 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap11 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap12 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap12 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap13 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap13 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap14 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap14 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap15 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap15 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap2 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap2 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap3 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap3 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap4 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap4 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap5 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap5 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap6 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap6 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap7 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap7 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap8 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap8 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderState.Wrap9 Property

Gets or sets the texture-wrapping behavior for multiple sets of texture coordinates. The default value for this render state is [TextureWrapCoordinates.Zero](#) (wrapping disabled in all directions).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureWrapCoordinates Wrap9 { get; set; }
```

Property Value

Combination of values from [TextureWrapCoordinates](#) to set or get.

Remarks

Valid values for this render state can be **Zero**, **One**, or both. These cause the system to wrap in the direction of the first, second, third, and fourth dimensions, sometimes referred to as the s, t, r, and q directions, for a given texture.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[RenderState Class](#)

[RenderState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderTarget Class

Represents a resource that will be written to at the end of a render pass. This is the base class for [RenderTarget2D](#) and [RenderTargetCube](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class RenderTarget : IDisposable
```

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTarget** objects must be recreated when the device is reset.

To use a **RenderTarget**, you must:

1. Create the **RenderTarget**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
```

```
        CheckTextureSize(device.PresentationParameters.BackBufferWidth,
            device.PresentationParameters.BackBufferHeight,
            out width, out height);

        // Create our render target
        return new RenderTarget2D(device,
            width, height, numberLevels, surface,
            type, 0);
    }
```

2. Set the **RenderTarget**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTarget**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTarget** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTarget**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

[What Is a Render Target?](#)

Tasks

[How To: Load Content](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[SetRenderTarget](#)

[RenderTarget2D](#)

[RenderTargetCube](#)

[GraphicsDeviceManager.DeviceReset Event](#)

[RenderTarget Members](#)












[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







RenderTarget Members

The following tables list the members exposed by the RenderTarget type.





Public Properties

Name	Description
 Format	Gets the format of the render target.
 GraphicsDevice	Gets the graphics device associated with this render target resource.
 Height	Gets the height, in pixels, of this render target.
 IsContentLost	Gets the current state of the content on a device.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 MultiSampleQuality	Gets the number of quality stops available for a given multisample type.
 MultiSampleType	Gets the levels of full-scene multisampling that the device can apply.
 Name	Gets the name of this render-target resource.
 RenderTargetUsage	Gets or sets render target usage flags.
 Tag	Gets the resource tags for this render target.
 Width	Gets the width, in pixels, of this render target.



Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs when content is about to be lost on a GraphicsDevice .
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 ContentLost	Occurs when resources are lost (for example, when the current device is lost).
 Disposing	Occurs when RenderTarget.Dispose Method is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also







Reference

[RenderTarget Class](#)





[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget Methods

Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs when content is about to be lost on a GraphicsDevice .
 raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[RenderTarget Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
RenderTarget.Dispose ()	Immediately releases the unmanaged resources used by this object.
RenderTarget.Dispose (Boolean)	Releases the unmanaged resources used by the RenderTarget and optionally releases the managed resources.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Dispose Method (Boolean)

Releases the unmanaged resources used by the [RenderTarget](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\] true](#) to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [RenderTarget](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.raise_ContentLost Method

Note

This method is available only when developing for Windows.

Occurs when content is about to be lost on a [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void raise_ContentLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[ContentLost](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderTarget.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[RenderTarget Class](#)












[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

RenderTarget Properties

Public Properties

	Name	Description
	Format	Gets the format of the render target.
	GraphicsDevice	Gets the graphics device associated with this render target resource.
	Height	Gets the height, in pixels, of this render target.
	IsContentLost	Gets the current state of the content on a device.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	MultiSampleQuality	Gets the number of quality stops available for a given multisample type.
	MultiSampleType	Gets the levels of full-scene multisampling that the device can apply.
	Name	Gets the name of this render-target resource.
	RenderTargetUsage	Gets or sets render target usage flags.
	Tag	Gets the resource tags for this render target.
	Width	Gets the width, in pixels, of this render target.

See Also

Reference

[RenderTarget Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget.Format Property

Gets the format of the render target.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; }
```

Property Value

The format of the render target.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.GraphicsDevice Property

Gets the graphics device associated with this render target resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The graphics device associated with this render target resource.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Height Property

Gets the height, in pixels, of this render target.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

The height, in pixels, of this render target.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.IsContentLost Property

Gets the current state of the content on a device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsContentLost { get; }
```

Property Value

true if content was lost due to device lost or similar event; otherwise **false**.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[ContentLost](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.MultiSampleQuality Property

Gets the number of quality stops available for a given multisample type.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MultiSampleQuality { get; }
```

Property Value

The number of quality stops available for a given multisample type.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.MultiSampleType Property

Gets the levels of full-scene multisampling that the device can apply.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MultiSampleType MultiSampleType { get; }
```

Property Value

The levels of full-scene multisampling that the device can apply.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Name Property

Gets the name of this render-target resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of this render-target resource.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.RenderTargetUsage Property

Gets or sets render target usage flags.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTargetUsage RenderTargetUsage { get; }
```

Property Value

Usage flags for the render target.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Tag Property

Gets the resource tags for this render target.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tags for this render target.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Width Property

Gets the width, in pixels, of this render target.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

The width, in pixels, of this render target.

See Also

Reference

[RenderTarget Class](#)



[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget Events

Public Events

	Name	Description
	Content Lost	Occurs when resources are lost (for example, when the current device is lost).
	Disposing	Occurs when RenderTarget.Dispose Method is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[RenderTarget Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget.ContentLost Event

Occurs when resources are lost (for example, when the current device is lost).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual event EventHandler ContentLost
```

Remarks

When **ContentLost** fires, [IsContentLost](#) is set to **true**. This property resets when the content is restored through a call to a [SetData](#) method or the rendering of a render target.

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[raise_ContentLost](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget.Disposing Event

Occurs when [RenderTarget.Dispose Method](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

Remarks [IsDisposed](#) indicates whether that object has been disposed.

Example

To add an event handler that listens for the **Disposing** event, use the following C# code.

```
obj.Disposing += new System.EventHandler( this.OnDisposing );
```

See Also

Reference

[RenderTarget Class](#)

[RenderTarget Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget2D Class

Represents a 2D texture resource that will be written to at the end of a render pass.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class RenderTarget2D : RenderTarget
```

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTarget2D** objects must be recreated when the device is reset.

To use a **RenderTarget2D**, you must:

1. Create the **RenderTarget2D**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
```

```
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

2. Set the **RenderTarget2D**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTarget2D**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTarget2D** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTarget2D**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

[What Is a Render Target?](#)

Tasks

[How To: Load Content](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[RenderTarget](#)

[GraphicsDeviceManager.DeviceReset Event](#)

[RenderTarget2D Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune












RenderTarget2D Members

The following tables list the members exposed by the RenderTarget2D type.








Public Constructors

Name	Description
 RenderTarget2D	Overloaded. Initializes a new instance of this class.




Public Properties

Name	Description
 Format	(Inherited from RenderTarget .)
 GraphicsDevice	(Inherited from RenderTarget .)
 Height	(Inherited from RenderTarget .)
 IsContentLost	(Inherited from RenderTarget .)
 IsDisposed	(Inherited from RenderTarget .)
 MultiSampleQuality	(Inherited from RenderTarget .)
 MultiSampleType	(Inherited from RenderTarget .)
 Name	(Inherited from RenderTarget .)
 RenderTargetUsage	(Inherited from RenderTarget .)
 Tag	(Inherited from RenderTarget .)
 Width	(Inherited from RenderTarget .)



Public Methods

Name	Description
 Dispose	(Inherited from RenderTarget .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetTexture	Gets the 2D texture associated with this render target.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	(Inherited from RenderTarget .)
 raise_Disposing	(Inherited from RenderTarget .)

Public Events

Name	Description
 ContentLost	(Inherited from RenderTarget .)
 Disposing	(Inherited from RenderTarget .)

See Also

Reference

[RenderTarget2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget2D Constructor

Initializes a new instance of this class.

Overload List

Name	Description
RenderTarget2D (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat)	Initializes a new instance of this class.
RenderTarget2D (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat, MultiSampleType, Int32)	Initializes a new instance of this class.
RenderTarget2D (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat, MultiSampleType, Int32, RenderTargetUsage)	Initializes a new instance of RenderTarget2D with the specified values.
RenderTarget2D (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat, RenderTargetUsage)	Initializes a new instance of RenderTarget2D with the specified values.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTarget2D** objects must be recreated when the device is reset.

To use a **RenderTarget2D**, you must:

1. Create the **RenderTarget2D**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
```

```
        CheckTextureSize(device.PresentationParameters.BackBufferWidth,
            device.PresentationParameters.BackBufferHeight,
            out width, out height);

        // Create our render target
        return new RenderTarget2D(device,
            width, height, numberLevels, surface,
            type, 0);
    }
```

2. Set the **RenderTarget2D**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTarget2D**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTarget2D** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTarget2D**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset Event](#)

[RenderTarget2D Class](#)

[RenderTarget2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget2D Constructor (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat)

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTarget2D (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    int numberLevels,
    SurfaceFormat format
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

width

Width, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferWidth** to get the current screen width.

height

Height, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferHeight** to get the current screen height.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If this is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures.

format

Surface format of the render target. Use [CheckDeviceFormat](#) to determine if a particular surface format is usable in a render target.

Exceptions

Exception type	Condition
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> <i>width</i> is larger than MaxTextureWidth. <i>height</i> is larger than MaxTextureHeight. The graphics device does not support creating a render target of type <i>format</i>. CheckDeviceFormat may be used to verify if a surface format is supported. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels.
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>width</i> or <i>height</i> is less than or equal to zero. <i>width</i> and <i>height</i> must be greater than zero.
InvalidOperationException	Unable to create this RenderTarget2D resource on the graphics device.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card.

Because of this, **RenderTarget2D** objects must be recreated when the device is reset.

To use a **RenderTarget2D**, you must:

1. Create the **RenderTarget2D**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

2. Set the **RenderTarget2D**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTarget2D**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTarget2D**. **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTarget2D**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Tasks

[How To: Load Content](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[RenderTarget2D](#) Class

[RenderTarget2D](#) Members

[CheckDeviceFormat](#)

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget2D Constructor (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat, MultiSampleType, Int32)

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTarget2D (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    int numberLevels,
    SurfaceFormat format,
    MultiSampleType multiSampleType,
    int multiSampleQuality
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

width

Width, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferWidth** to get the current screen width.

height

Height, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferHeight** to get the current screen height.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If this is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures.

format

Surface format of the render target. Use [CheckDeviceFormat](#) to determine if a particular surface format is usable in a render target.

multiSampleType

The levels of full-scene multisampling that the device can apply. You can use **graphicsDevice.PresentationParameters.MultiSampleType** to get the [MultiSampleType](#) of the back buffer.

multiSampleQuality

The number of quality stops available for a given multisample type. You can use **graphicsDevice.PresentationParameters.MultiSampleQuality** to get the multisample quality of the back buffer.

Exceptions

Exception type	Condition
ArgumentException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>width</i> is larger than MaxTextureWidth. <i>height</i> is larger than MaxTextureHeight. The graphics device does not support creating a render target of type <i>format</i>. CheckDeviceFormat may be used to verify if a surface format is supported. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels.
ArgumentNullException	<i>graphicsDevice</i> is null .

ArgumentOutOfRangeException	<i>width</i> or <i>height</i> is less than or equal to zero. <i>width</i> and <i>height</i> must be greater than zero.
InvalidOperationException	Unable to create this RenderTarget2D resource on the graphics device.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTarget2D** objects must be recreated when the device is reset.

To use a **RenderTarget2D**, you must:

1. Create the **RenderTarget2D**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

```
}
```

2. Set the **RenderTarget2D**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTarget2D**.

C#

```
// Render the shadow map  
GraphicsDevice.Clear(Color.Black);  
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTarget2D** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer  
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTarget2D**.

C#

```
// Return the shadow map as a texture  
return shadowRenderTarget.GetTexture();
```

See Also

Tasks

[How To: Load Content](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[GraphicsDeviceManager.DeviceReset Event](#)

[RenderTarget2D Class](#)

[RenderTarget2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget2D Constructor (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat, MultiSampleType, Int32, RenderTargetUsage)

Initializes a new instance of **RenderTarget2D** with the specified values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTarget2D (  
    GraphicsDevice graphicsDevice,  
    int width,  
    int height,  
    int numberLevels,  
    SurfaceFormat format,  
    MultiSampleType multiSampleType,  
    int multiSampleQuality,  
    RenderTargetUsage usage  
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

width

Width, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferWidth** to get the current screen width.

height

Height, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferHeight** to get the current screen height.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mipmap levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If this is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures.

format

Surface format of the render target. Use [CheckDeviceFormat](#) to determine if a particular surface format is usable in a render target.

multiSampleType

The levels of full-scene multisampling that the device can apply. You can use **graphicsDevice.PresentationParameters.MultiSampleType** to get the [MultiSampleType](#) of the back buffer.

multiSampleQuality

The number of quality stops available for a given multisample type. You can use **graphicsDevice.PresentationParameters.MultiSampleQuality** to get the multisample quality of the back buffer.

usage

Options identifying the behaviors of this render target resource.

See Also

Tasks

[How To: Load Content](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[RenderTarget2D Class](#)

[RenderTarget2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget2D Constructor (GraphicsDevice, Int32, Int32, Int32, SurfaceFormat, RenderTargetUsage)

Initializes a new instance of **RenderTarget2D** with the specified values.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTarget2D (  
    GraphicsDevice graphicsDevice,  
    int width,  
    int height,  
    int numberLevels,  
    SurfaceFormat format,  
    RenderTargetUsage usage  
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

width

Width, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferWidth** to get the current screen width.

height

Height, in pixels, of the render target. You can use **graphicsDevice.PresentationParameters.BackBufferHeight** to get the current screen height.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mipmap levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If this is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures.

format

Surface format of the render target. Use [CheckDeviceFormat](#) to determine if a particular surface format is usable in a render target.

usage

Options identifying the behaviors of this render target resource.

See Also

Tasks

[How To: Load Content](#)

[How To: Create a Depth Texture](#)

[How To: Implement Shadow Mapping](#)

Reference

[RenderTarget2D Class](#)








[RenderTarget2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget2D Methods

Public Methods

	Name	Description
	Dispose	(Inherited from RenderTarget .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetTexture	Gets the 2D texture associated with this render target.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_ContentLost	(Inherited from RenderTarget .)
	raise_Disposing	(Inherited from RenderTarget .)

See Also

Reference

[RenderTarget2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget2D.GetTexture Method

Gets the 2D texture associated with this render target.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetTexture ()
```

Return Value

The 2D texture associated with this render target.

Exceptions

Exception type	Condition
InvalidOperationException	A render target must be resolved (for example, setting the index to null or a different index) before calling GetTexture .

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewPort.Width,
    leftViewPort.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);

// TODO: Add your code to draw to the render target here.
// For example, this could be a call to Mesh.Draw,
// a SpriteBatch Begin-End sequence, or a call to
// DrawIndexedPrimitives, as you would have called them
// in the Draw method of your application.

// Set the device render target back to the back buffer.
graphics.GraphicsDevice.SetRenderTarget(0, null);

// Call GetTexture to retrieve the render target data and save it to a texture.
capturedTexture = renderTarget.GetTexture();
```

See Also

Reference

[RenderTarget2D Class](#)












[RenderTarget2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

RenderTarget2D Properties

Public Properties

	Name	Description
	Format	(Inherited from RenderTarget.)
	GraphicsDevice	(Inherited from RenderTarget.)
	Height	(Inherited from RenderTarget.)
	IsContentLost	(Inherited from RenderTarget.)
	IsDisposed	(Inherited from RenderTarget.)
	MultiSampleQuality	(Inherited from RenderTarget.)
	MultiSampleType	(Inherited from RenderTarget.)
	Name	(Inherited from RenderTarget.)
	RenderTargetUsage	(Inherited from RenderTarget.)
	Tag	(Inherited from RenderTarget.)
	Width	(Inherited from RenderTarget.)

See Also



Reference

[RenderTarget2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTarget2D Events

Public Events

	Name	Description
	ContentLost	(Inherited from RenderTarget.)
	Disposing	(Inherited from RenderTarget.)

See Also

Reference

[RenderTarget2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTargetCube Class

Represents a cubic texture resource that will be written to at the end of a render pass.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class RenderTargetCube : RenderTarget
```

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTargetCube** objects must be recreated when the device is reset.

To use a **RenderTargetCube**, you must:

1. Create the **RenderTargetCube**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
```

```
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

2. Set the **RenderTargetCube**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTargetCube**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTargetCube** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTargetCube**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

[What Is a Render Target?](#)

Tasks

[How To: Load Content](#)

Reference

[SetRenderTarget](#)

[GraphicsDeviceManager.DeviceReset Event](#)

[RenderTargetCube Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista












RenderTargetCube Members

The following tables list the members exposed by the RenderTargetCube type.








Public Constructors

Name	Description
 RenderTargetCube	Overloaded. Initializes a new instance of this class.




Public Properties

Name	Description
 Format	(Inherited from RenderTarget .)
 GraphicsDevice	(Inherited from RenderTarget .)
 Height	(Inherited from RenderTarget .)
 IsContentLost	(Inherited from RenderTarget .)
 IsDisposed	(Inherited from RenderTarget .)
 MultiSampleQuality	(Inherited from RenderTarget .)
 MultiSampleType	(Inherited from RenderTarget .)
 Name	(Inherited from RenderTarget .)
 RenderTargetUsage	(Inherited from RenderTarget .)
 Tag	(Inherited from RenderTarget .)
 Width	(Inherited from RenderTarget .)



Public Methods

Name	Description
 Dispose	(Inherited from RenderTarget .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetTexture	Gets a copy of the cube texture associated with this render target.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	(Inherited from RenderTarget .)
 raise_Disposing	(Inherited from RenderTarget .)

Public Events

Name	Description
 ContentLost	(Inherited from RenderTarget .)
 Disposing	(Inherited from RenderTarget .)

See Also

Reference

[RenderTargetCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTargetCube Constructor

Initializes a new instance of this class.

Overload List

Name	Description
RenderTargetCube (GraphicsDevice, Int32, Int32, SurfaceFormat)	Initializes a new instance of this class.
RenderTargetCube (GraphicsDevice, Int32, Int32, SurfaceFormat, MultiSampleType, Int32)	Initializes a new instance of this class.
RenderTargetCube (GraphicsDevice, Int32, Int32, SurfaceFormat, MultiSampleType, Int32, RenderTargetUsage)	Initializes a new instance of RenderTargetCube .
RenderTargetCube (GraphicsDevice, Int32, Int32, SurfaceFormat, RenderTargetUsage)	Initializes a new instance of this class.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTargetCube** objects must be recreated when the device is reset.

To use a **RenderTargetCube**, you must:

1. Create the **RenderTargetCube**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
```



```
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

2. Set the **RenderTargetCube**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTargetCube**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTargetCube** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTargetCube**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[RenderTargetCube](#) Class

[RenderTargetCube](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

RenderTargetCube Constructor (GraphicsDevice, Int32, Int32, SurfaceFormat)

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTargetCube (
    GraphicsDevice graphicsDevice,
    int size,
    int numberLevels,
    SurfaceFormat format
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

size

The width and height of this cube texture resource, in pixels.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If this is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures.

format

Surface format of the render target.

Exceptions

Exception type	Condition
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> The graphics device does not support creating a render target of type <i>format</i>. <i>size</i> is larger than MaxTextureWidth or MaxTextureHeight. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels.
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>size</i> is less than or equal to zero. <i>size</i> must be greater than zero.
InvalidOperationException	Unable to create this RenderTargetCube resource on the graphics device.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTargetCube** objects must be recreated when the device is reset.

To use a **RenderTargetCube**, you must:

1. Create the **RenderTargetCube**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

2. Set the **RenderTargetCube**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTargetCube**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTargetCube** **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTargetCube**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Reference

[RenderTargetCube Class](#)

[RenderTargetCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderTargetCube Constructor (GraphicsDevice, Int32, Int32, SurfaceFormat, MultiSampleType, Int32)

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTargetCube (
    GraphicsDevice graphicsDevice,
    int size,
    int numberLevels,
    SurfaceFormat format,
    MultiSampleType multiSampleType,
    int multiSampleQuality
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

size

The width and height of this cube texture resource, in pixels.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

format

Surface format of the render target.

multiSampleType

The levels of full-scene multisampling that the device can apply.

multiSampleQuality

The number of quality stops available for a given multisample type.

Exceptions

Exception type	Condition
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> The graphics device does not support creating a render target of type <i>format</i>. <i>size</i> is larger than MaxTextureWidth or MaxTextureHeight. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels.
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>size</i> is less than or equal to zero. <i>size</i> must be greater than zero.
InvalidOperationException	Unable to create this RenderTargetCube resource on the graphics device.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTargetCube** objects must be recreated when the device is reset.

To use a **RenderTargetCube**, you must:

1. Create the **RenderTargetCube**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,  
    1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,  
    int numberLevels, SurfaceFormat surface)  
{  
    MultiSampleType type =  
        device.PresentationParameters.MultiSampleType;  
  
    // If the card can't use the surface format  
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(  
        DeviceType.Hardware,  
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,  
        TextureUsage.None,  
        QueryUsages.None,  
        ResourceType.RenderTarget,  
        surface))  
    {  
        // Fall back to current display format  
        surface = device.DisplayMode.Format;  
    }  
    // Or it can't accept that surface format  
    // with the current AA settings  
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(  
        DeviceType.Hardware, surface,  
        device.PresentationParameters.IsFullScreen, type))  
    {  
        // Fall back to no antialiasing  
        type = MultiSampleType.None;  
    }  
  
    int width, height;  
  
    // See if we can use our buffer size as our texture  
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,  
        device.PresentationParameters.BackBufferHeight,  
        out width, out height);  
  
    // Create our render target  
    return new RenderTarget2D(device,  
        width, height, numberLevels, surface,  
        type, 0);  
}
```

2. Set the **RenderTargetCube**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTargetCube**.

C#

```
// Render the shadow map  
GraphicsDevice.Clear(Color.Black);
```

```
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTargetCube**. **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer  
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTargetCube**.

C#

```
// Return the shadow map as a texture  
return shadowRenderTarget.GetTexture();
```

See Also

Reference

[RenderTargetCube Class](#)

[RenderTargetCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderTargetCube Constructor (GraphicsDevice, Int32, Int32, SurfaceFormat, MultiSampleType, Int32, RenderTargetUsage)

Initializes a new instance of **RenderTargetCube**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTargetCube (  
    GraphicsDevice graphicsDevice,  
    int size,  
    int numberLevels,  
    SurfaceFormat format,  
    MultiSampleType multiSampleType,  
    int multiSampleQuality,  
    RenderTargetUsage usage  
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

size

The width and height of this cube texture resource, in pixels.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

format

Surface format of the render target.

multiSampleType

Levels of full-scene multisampling that the device can apply.

multiSampleQuality

Number of quality stops available for a given multisample type.

usage

Options identifying the behaviors of this texture resource.

See Also

Reference

[RenderTargetCube Class](#)

[RenderTargetCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderTargetCube Constructor (GraphicsDevice, Int32, Int32, SurfaceFormat, RenderTargetUsage)

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public RenderTargetCube (
    GraphicsDevice graphicsDevice,
    int size,
    int numberLevels,
    SurfaceFormat format,
    RenderTargetUsage usage
)
```

Parameters

graphicsDevice

The graphics device to associate with this render target resource.

size

The width and height of this cube texture resource, in pixels.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

format

Surface format of the render target.

usage

Options identifying the behaviors of this texture resource.

Exceptions

Exception type	Condition
ArgumentException	One of the following conditions is true: <ul style="list-style-type: none"> The graphics device does not support creating a render target of type <i>format</i>. <i>size</i> is larger than MaxTextureWidth or MaxTextureHeight. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels.
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>size</i> is less than or equal to zero. <i>size</i> must be greater than zero.
InvalidOperationException	Unable to create this RenderTargetCube resource on the graphics device.

Remarks

After a render pass the render target contains the color information of a rendered image.

Render targets represent a linear area of display memory and usually reside in the display memory of the display card. Because of this, **RenderTargetCube** objects must be recreated when the device is reset.

To use a **RenderTargetCube**, you must:

1. Create the **RenderTargetCube**.

C#

```
shadowRenderTarget = GfxComponent.CreateRenderTarget(GraphicsDevice,
```

```
1, SurfaceFormat.Single);
```

C#

```
public static RenderTarget2D CreateRenderTarget(GraphicsDevice device,
    int numberLevels, SurfaceFormat surface)
{
    MultiSampleType type =
        device.PresentationParameters.MultiSampleType;

    // If the card can't use the surface format
    if (!GraphicsAdapter.DefaultAdapter.CheckDeviceFormat(
        DeviceType.Hardware,
        GraphicsAdapter.DefaultAdapter.CurrentDisplayMode.Format,
        TextureUsage.None,
        QueryUsages.None,
        ResourceType.RenderTarget,
        surface))
    {
        // Fall back to current display format
        surface = device.DisplayMode.Format;
    }
    // Or it can't accept that surface format
    // with the current AA settings
    else if (!GraphicsAdapter.DefaultAdapter.CheckDeviceMultiSampleType(
        DeviceType.Hardware, surface,
        device.PresentationParameters.IsFullScreen, type))
    {
        // Fall back to no antialiasing
        type = MultiSampleType.None;
    }

    int width, height;

    // See if we can use our buffer size as our texture
    CheckTextureSize(device.PresentationParameters.BackBufferWidth,
        device.PresentationParameters.BackBufferHeight,
        out width, out height);

    // Create our render target
    return new RenderTarget2D(device,
        width, height, numberLevels, surface,
        type, 0);
}
```

2. Set the **RenderTargetCube**.

C#

```
GraphicsDevice.SetRenderTarget(0, shadowRenderTarget);
```

3. Draw into the **RenderTargetCube**.

C#

```
// Render the shadow map
GraphicsDevice.Clear(Color.Black);
DrawScene(MyEffect.shadowMap);
```

4. Reset the original **RenderTargetCube**. **null** as the default render target - i.e. the back buffer.

C#

```
// Set render target back to the back buffer
GraphicsDevice.SetRenderTarget(0, null);
```

5. Call [GetTexture](#) to retrieve the contents of the **RenderTargetCube**.

C#

```
// Return the shadow map as a texture
return shadowRenderTarget.GetTexture();
```

See Also

Reference

[RenderTargetCube Class](#)








[RenderTargetCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderTargetCube Methods

Public Methods

	Name	Description
	Dispose	(Inherited from RenderTarget .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetTexture	Gets a copy of the cube texture associated with this render target.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_ContentLost	(Inherited from RenderTarget .)
	raise_Disposing	(Inherited from RenderTarget .)

See Also

Reference

[RenderTargetCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTargetCube.GetTexture Method

Gets a copy of the cube texture associated with this render target.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCube GetTexture ()
```

Return Value

A copy of the cube texture associated with this render target.

Exceptions

Exception type	Condition
InvalidOperationException	A render target must be resolved (e.g. setting the index to null or a different index) before calling GetTexture .

Example

C#

```
// Create a new render target to receive the drawn data.
renderTarget = new RenderTarget2D(
    graphics.GraphicsDevice,
    // In this case, we are creating
    // a render target to match half of a split screen:
    leftViewPort.Width,
    leftViewPort.Height,
    // Number of levels in the render target:
    1,
    // Use the same surface format as the back buffer.
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

// Set the render target on the device.
graphics.GraphicsDevice.SetRenderTarget(0, renderTarget);

// TODO: Add your code to draw to the render target here.
// For example, this could be a call to Mesh.Draw,
// a SpriteBatch Begin-End sequence, or a call to
// DrawIndexedPrimitives, as you would have called them
// in the Draw method of your application.

// Set the device render target back to the back buffer.
graphics.GraphicsDevice.SetRenderTarget(0, null);

// Call GetTexture to retrieve the render target data and save it to a texture.
capturedTexture = renderTarget.GetTexture();
```

See Also

Reference

[RenderTargetCube Class](#)












[RenderTargetCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

RenderTargetCube Properties

Public Properties

	Name	Description
	Format	(Inherited from RenderTarget .)
	GraphicsDevice	(Inherited from RenderTarget .)
	Height	(Inherited from RenderTarget .)
	IsContentLost	(Inherited from RenderTarget .)
	IsDisposed	(Inherited from RenderTarget .)
	MultiSampleQuality	(Inherited from RenderTarget .)
	MultiSampleType	(Inherited from RenderTarget .)
	Name	(Inherited from RenderTarget .)
	RenderTargetUsage	(Inherited from RenderTarget .)
	Tag	(Inherited from RenderTarget .)
	Width	(Inherited from RenderTarget .)

See Also



Reference

[RenderTargetCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTargetCube Events

Public Events

	Name	Description
	ContentLost	(Inherited from RenderTarget.)
	Disposing	(Inherited from RenderTarget.)

See Also

Reference

[RenderTargetCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

RenderTargetUsage Enumeration

Determines how render target data is used once a new render target is set.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum RenderTargetUsage
```

Members

Member name	Description
DiscardContents	Always clears the render target data.
PlatformContents	Either clears or keeps the data, depending on the current platform. On Xbox 360, the render target will discard contents. On PC, the render target will discard if multisampling is enabled, and preserve the contents if not.
PreserveContents	Always keeps the render target data.

See Also

Concepts

[What Is a Render Target?](#)

Reference

[SetRenderTarget](#)

[GetTexture](#)

[GetTexture](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResolveTexture2D Class

Represents a 2D grid of texels. Use this class as a resolve target for a back buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class ResolveTexture2D : Texture2D
```

See Also

Reference

[ResolveTexture2D Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune














ResolveTexture2D Members

The following tables list the members exposed by the ResolveTexture2D type.














Public Constructors

Name	Description
 ResolveTexture2D	Initializes an empty instance of ResolveTexture2D.




Public Properties

Name	Description
 Format	(Inherited from Texture2D .)
 GraphicsDevice	(Inherited from GraphicsResource .)
 Height	(Inherited from Texture2D .)
 IsContentLost	Determines if the render target data has been lost due to a lost device event.
 IsDisposed	(Inherited from GraphicsResource .)
 LevelCount	(Inherited from Texture .)
 LevelOfDetail	(Inherited from Texture .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 Tag	(Inherited from GraphicsResource .)
 TextureUsage	(Inherited from Texture2D .)
 Width	(Inherited from Texture2D .)



Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 FromFile	(Inherited from Texture .)
 GenerateMipMaps	(Inherited from Texture .)
 GetCreationParameters	(Inherited from Texture .)
 GetData	(Inherited from Texture2D .)
 GetHashCode	(Inherited from Object .)
 GetTextureInformation	(Inherited from Texture .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Save	(Inherited from Texture .)
 SetData	(Inherited from Texture2D .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 ContentLost	Occurs when the render target data is lost due to a lost device event.
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[ResolveTexture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResolveTexture2D Constructor

Initializes an empty instance of **ResolveTexture2D**.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ResolveTexture2D (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    int numberLevels,
    SurfaceFormat format
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) used to display the texture.

width

The width of the texture, in pixels. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. If this value is 0, *width* is set to 1.

height

The height of the texture, in pixels. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. If this value is 0, *height* is set to 1.

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If *numberLevels* is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures. Use [LevelCount](#) to see the number of levels generated.

format

A [SurfaceFormat](#) value specifying the requested pixel format for the texture. The returned texture may be of a different format if the device does not support the requested format. Applications should check the format of the returned texture to ensure that it matches the requested format.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>width</i> is larger than MaxTextureWidth. <i>height</i> is larger than MaxTextureHeight. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <p>The parameters specified are not compatible:</p> <ul style="list-style-type: none"> <i>format</i> and <i>numberLevels</i> are incompatible. Multi-element textures of type SurfaceFormat.Multi2Bgra32 cannot be mipmapped, set <i>numberLevels</i> to 1. The <i>format</i> requested is not supported by the graphics device. The device does not support creating a texture of the given format with the given TextureUsage.
ArgumentOutOfRangeException	<i>width</i> or <i>height</i> is less than or equal to zero; <i>width</i> and <i>height</i> must be greater than zero.

OutOfVideoMemoryException	Unable to create this resource on the graphics device.
---	--

See Also

Reference

[ResolveTexture2D Class](#)














[ResolveTexture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResolveTexture2D Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	FromFile	(Inherited from Texture .)
	GenerateMipMaps	(Inherited from Texture .)
	GetCreationParameters	(Inherited from Texture .)
	GetData	(Inherited from Texture2D .)
	GetHashCode	(Inherited from Object .)
	GetTextureInformation	(Inherited from Texture .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	(Inherited from Texture .)
	SetData	(Inherited from Texture2D .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_ContentLost	Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[ResolveTexture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResolveTexture2D.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
ResolveTexture2D.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.
ResolveTexture2D.Dispose ()	(Inherited from GraphicsResource .)

See Also

Reference

[ResolveTexture2D Class](#)

[ResolveTexture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResolveTexture2D.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[ResolveTexture2D Class](#)

[ResolveTexture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResolveTexture2D.raise_ContentLost Method

Note

This method is available only when developing for Windows.

Occurs after content is lost from a graphics device failure, allowing an application to re-create all resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void raise_ContentLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[ResolveTexture2D Class](#)














[ResolveTexture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ResolveTexture2D Properties

Public Properties

	Name	Description
	Format	(Inherited from Texture2D .)
	GraphicsDevice	(Inherited from GraphicsResource .)
	Height	(Inherited from Texture2D .)
	IsContentLost	Determines if the render target data has been lost due to a lost device event.
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	(Inherited from Texture .)
	LevelOfDetail	(Inherited from Texture .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Tag	(Inherited from GraphicsResource .)
	TextureUsage	(Inherited from Texture2D .)
	Width	(Inherited from Texture2D .)

See Also

Reference

[ResolveTexture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResolveTexture2D.IsContentLost Property

Determines if the render target data has been lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsContentLost { get; }
```

Property Value

true if the content was lost; **false** otherwise.

See Also

Reference

[ResolveTexture2D Class](#)



[ResolveTexture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResolveTexture2D Events

Public Events

	Name	Description
	ContentLost	Occurs when the render target data is lost due to a lost device event.
	Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[ResolveTexture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResolveTexture2D.ContentLost Event

Occurs when the render target data is lost due to a lost device event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual event EventHandler ContentLost
```

See Also

Reference

[ResolveTexture2D Class](#)

[ResolveTexture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceCreatedEventArgs Class

Contains event data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ResourceCreatedEventArgs : EventArgs
```

See Also

Reference

[ResourceCreatedEventArgs Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






ResourceCreatedEventArgs Members

The following tables list the members exposed by the ResourceCreatedEventArgs type.



Public Properties

	Name	Description
	Resource	The object raising the event.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ResourceCreatedEventArgs Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ResourceCreatedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[ResourceCreatedEventArgs Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResourceCreatedEventArgs Properties

Public Properties

	Name	Description
	Resource	The object raising the event.

See Also

Reference

[ResourceCreatedEventArgs Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResourceCreatedEventArgs.Resource Property

The object raising the event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Resource { get; }
```

Property Value

The object raising the event.

See Also

Reference

[ResourceCreatedEventArgs Class](#)

[ResourceCreatedEventArgs Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceDestroyedEventArgs Class

Arguments for a [ResourceDestroyed](#) event.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ResourceDestroyedEventArgs : EventArgs
```

See Also

Reference

[ResourceDestroyedEventArgs Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






ResourceDestroyedEventArgs Members

The following tables list the members exposed by the ResourceDestroyedEventArgs type.



Public Properties

	Name	Description
	Name	Gets the name of the destroyed resource.
	Tag	Gets the resource manager tag of the destroyed resource.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ResourceDestroyedEventArgs Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ResourceDestroyedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[ResourceDestroyedEventArgs Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResourceDestroyedEventArgs Properties

Public Properties

	Name	Description
	Name	Gets the name of the destroyed resource.
	Tag	Gets the resource manager tag of the destroyed resource.

See Also

Reference

[ResourceDestroyedEventArgs Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ResourceDestroyedEventArgs.Name Property

Gets the name of the destroyed resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of the destroyed resource.

See Also

Reference

[ResourceDestroyedEventArgs Class](#)

[ResourceDestroyedEventArgs Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceDestroyedEventArgs.Tag Property

Gets the resource manager tag of the destroyed resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; }
```

Property Value

The resource manager tag of the destroyed resource.

See Also

Reference

[ResourceDestroyedEventArgs Class](#)

[ResourceDestroyedEventArgs Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ResourceType Enumeration

Defines resource types.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ResourceType
```

Members

Member name	Description
DepthStencilBuffer	A depth stencil buffer resource.
IndexBuffer	An index buffer resource.
RenderTarget	A render target resource.
Texture2D	A 2-dimensional texture resource.
Texture3D	A 3-dimensional texture resource.
Texture3DVolume	A 3-D volume texture resource.
TextureCube	A cube texture resource.
VertexBuffer	A vertex buffer resource.

See Also

Reference

[GraphicsAdapter.CheckDeviceFormat Method](#)

[GraphicsResource.ResourceType Property](#)

[TextureInformation.ResourceType Property](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SamplerState Class

Contains sampler states for the device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SamplerState
```

See Also

Reference

[SamplerState Members](#)











[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






SamplerState Members

The following tables list the members exposed by the SamplerState type.



Public Properties

Name	Description
 AddressU	Gets or sets the texture-address mode for the u-coordinate.
 AddressV	Gets or sets the texture-address mode for the v-coordinate.
 AddressW	Gets or sets the texture-address mode for the w-coordinate.
 BorderColor	Gets or sets the border color.
 MagFilter	Gets or sets a magnification filter.
 MaxAnisotropy	Gets or sets the maximum anisotropy. The default value is 0.
 MaxMipLevel	Gets or sets the level of detail (LOD) index of the largest map to use.
 MinFilter	Gets or sets a minification filter.
 MipFilter	Gets or sets a mipmap filter to use during minification.
 MipMapLevelOfDetailBias	Gets or sets the mipmap level of detail (LOD) bias. The default value is 0.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[SamplerState Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

SamplerState Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SamplerState Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SamplerState.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[SamplerState Class](#)











[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState Properties

Public Properties

	Name	Description
	AddressU	Gets or sets the texture-address mode for the u-coordinate.
	AddressV	Gets or sets the texture-address mode for the v-coordinate.
	AddressW	Gets or sets the texture-address mode for the w-coordinate.
	BorderColor	Gets or sets the border color.
	MagFilter	Gets or sets a magnification filter.
	MaxAnisotropy	Gets or sets the maximum anisotropy. The default value is 0.
	MaxMipLevel	Gets or sets the level of detail (LOD) index of the largest map to use.
	MinFilter	Gets or sets a minification filter.
	MipFilter	Gets or sets a mipmap filter to use during minification.
	MipMapLevelOfDetailBias	Gets or sets the mipmap level of detail (LOD) bias. The default value is 0.

See Also

Reference

[SamplerState Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SamplerState.AddressU Property

Gets or sets the texture-address mode for the u-coordinate.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureAddressMode AddressU { get; set; }
```

Property Value

Texture-address mode to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.AddressV Property

Gets or sets the texture-address mode for the v-coordinate.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureAddressMode AddressV { get; set; }
```

Property Value

Texture-address mode to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.AddressW Property

Gets or sets the texture-address mode for the w-coordinate.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureAddressMode AddressW { get; set; }
```

Property Value

Texture-address mode to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.BorderColor Property

Gets or sets the border color.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color BorderColor { get; set; }
```

Property Value

Border color to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.MagFilter Property

Gets or sets a magnification filter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureFilter MagFilter { get; set; }
```

Property Value

The magnification filter to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.MaxAnisotropy Property

Gets or sets the maximum anisotropy. The default value is 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxAnisotropy { get; set; }
```

Property Value

The maximum anisotropy value to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.MaxMipLevel Property

Gets or sets the level of detail (LOD) index of the largest map to use.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxMipLevel { get; set; }
```

Property Value

The maximum LOD to set or get. This index value can range from 0 to $(n - 1)$, where n is the largest map.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.MinFilter Property

Gets or sets a minification filter.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureFilter MinFilter { get; set; }
```

Property Value

The minification filter to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.MipFilter Property

Gets or sets a mipmap filter to use during minification.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureFilter MipFilter { get; set; }
```

Property Value

The minification filter to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerState.MipMapLevelOfDetailBias Property

Gets or sets the mipmap level of detail (LOD) bias. The default value is 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MipMapLevelOfDetailBias { get; set; }
```

Property Value

The mipmap bias to set or get.

See Also

Reference

[SamplerState Class](#)

[SamplerState Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SamplerStateCollection Class

Collection of [SamplerState](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SamplerStateCollection
```

See Also

Reference

[SamplerStateCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






SamplerStateCollection Members

The following tables list the members exposed by the SamplerStateCollection type.



Public Properties

	Name	Description
	Item	Gets a specific SamplerState object using an index value.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[SamplerStateCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

SamplerStateCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[SamplerStateCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SamplerStateCollection Properties

Public Properties

	Name	Description
	Item	Gets a specific SamplerState object using an index value.

See Also

Reference

[SamplerStateCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SamplerStateCollection.Item Property

Gets a specific [SamplerState](#) object using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SamplerState this [
    int index
] { get; }
```

Property Value

The [SamplerState](#) object at the requested index.

Exceptions

Exception type	Condition
InvalidOperationException	The index for this vertex stream is invalid.

See Also

Reference

[SamplerStateCollection Class](#)

[SamplerStateCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SaveStateMode Enumeration

Defines options for saving the graphics device state before and after an effect technique is applied.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum SaveStateMode
```

Members

Member name	Description
None	No device state is saved when calling Begin or restored when calling End .
SaveState	Save all device state when calling Begin and restore device state when calling End .

See Also

Reference

[Effect.Begin Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SetDataOptions Enumeration

Describes whether existing buffer data will be overwritten or discarded during a **SetData** operation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum SetDataOptions
```

Members

Member name	Description
Discard	<p>The SetData operation will discard all existing buffer data.</p> <p>The operation will overwrite (with a write-only operation) every location within the locked surface. This is a valid option when using dynamic textures, dynamic vertex buffers, and dynamic index buffers. You may not use this option to update a portion of a surface.</p> <p>For vertex and index buffers, the SetData operation will discard the entire buffer. A pointer to a new memory area is returned so that the direct memory access (DMA) and rendering from the previous area do not stall.</p> <p>For textures, the SetData operation will overwrite (with a write-only operation) every location within the region being locked.</p>
None	Portions of existing data in the buffer may be overwritten during this operation.
NoOverwrite	The SetData operation will not overwrite existing data in the vertex and index buffers. Specifying this option allows the driver to return immediately from a SetData operation and continue rendering.

Remarks

Note

Discard is not valid on Xbox 360, but dynamic geometry may be rendered on the Xbox 360 by using [DrawUserPrimitives](#).

See Also

Reference

[GraphicsDevice.DrawUserPrimitives Generic Method](#)

[IndexBuffer.SetData Method](#)

[Texture2D.SetData Method](#)

[Texture3D.SetData Method](#)

[TextureCube.SetData Method](#)

[VertexBuffer.SetData Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ShaderCompiler Class

Note

This class is available only when developing for Windows.

Compiles and decompiles high-level shader language (HLSL) shaders.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class ShaderCompiler
```

See Also

Reference

[ShaderCompiler Members](#)
















[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



ShaderCompiler Members

The following tables list the members exposed by the ShaderCompiler type.

Public Methods

	Name	Description
 	AssembleFromFile	Overloaded. Compiles a shader from a file or stream containing shader assembly code (ASM).
 	AssembleFromSource	Compiles a shader from a string containing shader assembly code (ASM).
 	CompileFromFile	Overloaded. Compiles a shader from a file or stream containing shader source code.
 	CompileFromSource	Compiles a shader from a string containing the shader source code.
 	Disassemble	Disassembles a shader.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also
















Reference

[ShaderCompiler Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderCompiler Methods

Public Methods

	Name	Description
 	AssembleFromFile	Overloaded. Compiles a shader from a file or stream containing shader assembly code (ASM).
 	AssembleFromSource	Compiles a shader from a string containing shader assembly code (ASM).
 	CompileFromFile	Overloaded. Compiles a shader from a file or stream containing shader source code.
 	CompileFromSource	Compiles a shader from a string containing the shader source code.
 	Disassemble	Disassembles a shader.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ShaderCompiler Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderCompiler.AssembleFromFile Method

Compiles a shader from a file or stream containing shader assembly code (ASM).

Overload List

Name	Description
ShaderCompiler.AssembleFromFile (Stream, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)	Compiles a shader from a stream containing shader assembly code (ASM).
ShaderCompiler.AssembleFromFile (Stream, Int32, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)	Compiles a shader from a stream containing shader assembly code (ASM), specifying the number of bytes in the stream.
ShaderCompiler.AssembleFromFile (String, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)	Compiles a shader from a file containing shader assembly code (ASM).

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderCompiler.AssembleFromFile Method (Stream, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles a shader from a stream containing shader assembly code (ASM).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader AssembleFromFile (
    Stream shaderSourceCode,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    TargetPlatform platform
)
```

Parameters

shaderSourceCode

Stream that contains the shader data.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

Target platform for the compilation.

Return Value

The compiled shader.

Exceptions

Exception type	Condition
ArgumentNullException	<i>shaderSourceCode</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>shaderSourceCode</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>shaderSourceCode</i> does not contain enough data to support this call.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.AssembleFromFile Method (Stream, Int32, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles a shader from a stream containing shader assembly code (ASM), specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader AssembleFromFile (
    Stream shaderSourceCode,
    int numberBytes,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    TargetPlatform platform
)
```

Parameters

shaderSourceCode

Stream that contains the shader data.

numberBytes

The number of bytes in *shaderSourceCode*.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

Target platform for the compilation.

Return Value

The compiled shader.

Exceptions

Exception type	Condition
ArgumentNullException	<i>shaderSourceCode</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>shaderSourceCode</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>shaderSourceCode</i> does not contain enough data to support this call.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.AssembleFromFile Method (String, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles a shader from a file containing shader assembly code (ASM).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader AssembleFromFile (
    string shaderFileName,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    TargetPlatform platform
)
```

Parameters

shaderFileName

Name of the file containing the compiled shader data.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

Target platform for the compilation.

Return Value

The compiled shader.

Exceptions

Exception type	Condition
ArgumentException	<i>shaderFileName</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>shaderFileName</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>shaderFileName</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>shaderFileName</i> was not found.
NotSupportedException	<i>shaderFileName</i> is in an invalid format.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.AssembleFromSource Method

Note

This method is available only when developing for Windows.

Compiles a shader from a string containing shader assembly code (ASM).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader AssembleFromSource (  
    string shaderSourceCode,  
    CompilerMacro[] preprocessorDefines,  
    CompilerIncludeHandler includeHandler,  
    CompilerOptions options,  
    TargetPlatform platform  
)
```

Parameters

shaderSourceCode

String that contains the shader data.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

platform

Target platform for compilation.

Return Value

The compiled shader.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.CompileFromFile Method

Compiles a shader from a file or stream containing shader source code.

Overload List

Name	Description
ShaderCompiler.CompileFromFile (Stream, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, String, ShaderProfile, TargetPlatform)	Compiles a shader from a stream containing the shader source code.
ShaderCompiler.CompileFromFile (Stream, Int32, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, String, ShaderProfile, TargetPlatform)	Compiles a shader from a stream containing the shader source code, specifying the number of bytes in the stream.
ShaderCompiler.CompileFromFile (String, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, String, ShaderProfile, TargetPlatform)	Compiles a shader from file containing the shader source code.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderCompiler.CompileFromFile Method (Stream, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, String, ShaderProfile, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles a shader from a stream containing the shader source code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader CompileFromFile (
    Stream shaderSourceCode,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    string functionName,
    ShaderProfile profile,
    TargetPlatform platform
)
```

Parameters

shaderSourceCode

Stream that contains the shader source code.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

functionName

The function to be compiled.

profile

A shader profile that determines the shader instruction set.

platform

The target platform for compilation.

Return Value

The compiled shader.

Exceptions

Exception type	Condition
ArgumentNullException	<i>shaderSourceCode</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>shaderSourceCode</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>shaderSourceCode</i> does not contain enough data to support this call.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.CompileFromFile Method (Stream, Int32, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, String, ShaderProfile, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles a shader from a stream containing the shader source code, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader CompileFromFile (
    Stream shaderSourceCode,
    int numberBytes,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    string functionName,
    ShaderProfile profile,
    TargetPlatform platform
)
```

Parameters

shaderSourceCode

Stream that contains the shader source code.

numberBytes

The number of bytes in *shaderSourceCode*.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

functionName

The function to be compiled.

profile

A shader profile that determines the shader instruction set.

platform

The target platform for compilation.

Return Value

The compiled shader.

Exceptions

Exception type	Condition
ArgumentNullException	<i>shaderSourceCode</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>shaderSourceCode</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>shaderSourceCode</i> does not contain enough data to support this call.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.CompileFromFile Method (String, CompilerMacro[], CompilerIncludeHandler, CompilerOptions, String, ShaderProfile, TargetPlatform)

Note

This method is available only when developing for Windows.

Compiles a shader from file containing the shader source code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader CompileFromFile (
    string shaderFileName,
    CompilerMacro[] preprocessorDefines,
    CompilerIncludeHandler includeHandler,
    CompilerOptions options,
    string functionName,
    ShaderProfile profile,
    TargetPlatform platform
)
```

Parameters

shaderFileName

Name of the file containing the shader source code.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

functionName

The function to be compiled.

profile

A shader profile that determines the shader instruction set.

platform

The target platform for compilation.

Return Value

The compiled shader.

Exceptions

Exception type	Condition
ArgumentException	<i>shaderFileName</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>shaderFileName</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>shaderFileName</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>shaderFileName</i> was not found.
NotSupportedException	<i>shaderFileName</i> is in an invalid format.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.CompileFromSource Method

Note

This method is available only when developing for Windows.

Compiles a shader from a string containing the shader source code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static CompiledShader CompileFromSource (  
    string shaderSourceCode,  
    CompilerMacro[] preprocessorDefines,  
    CompilerIncludeHandler includeHandler,  
    CompilerOptions options,  
    string functionName,  
    ShaderProfile profile,  
    TargetPlatform platform  
)
```

Parameters

shaderSourceCode

String that contains the shader source code.

preprocessorDefines

Describes preprocessor definitions used by an effect object.

includeHandler

User-implemented interface to provide callbacks for **#include** directives during shader compilation.

options

Compilation optimization options.

functionName

The function to be compiled.

profile

A shader profile that determines the shader instruction set.

platform

Target platform for compilation.

Return Value

The compiled shader.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderCompiler.Disassemble Method

Note

This method is available only when developing for Windows.

Disassembles a shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static string Disassemble (  
    byte[] shaderCode,  
    bool enableColorCode,  
    string comments  
)
```

Parameters

shaderCode

The shader byte code to disassemble.

enableColorCode

[[MarshalAsAttribute](#)(U1)] Specifies whether to enable color coding to make the disassembly easier to read.

comments

A comment string to include at the top of the shader.

Return Value

The disassembled shader.

See Also

Reference

[ShaderCompiler Class](#)

[ShaderCompiler Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderConstant Class

Describes a shader constant.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ShaderConstant
```

See Also

Reference

[ShaderConstant Members](#)












[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








ShaderConstant Members

The following tables list the members exposed by the ShaderConstant type.



Public Properties

Name	Description
 ColumnCount	Gets the number of columns in the ShaderConstant .
 ElementCount	Gets the number of elements in the ShaderConstant .
 Name	Gets the name of the ShaderConstant .
 ParameterClass	Gets the EffectParameterClass of the ShaderConstant .
 ParameterType	Gets the EffectParameterType of the ShaderConstant .
 RegisterCount	Gets the number of registers that contain data.
 RegisterIndex	Gets the index of the ShaderConstant in the constant table.
 RegisterSet	Gets the data type of the shader register.
 RowCount	Gets the number of rows in the ShaderConstant .
 SamplerIndex	Gets the sampler index number of the ShaderConstant from the constant table.
 StructureMemberCount	Gets the number of structure member subparameters.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetValue	Overloaded. Sets the value of a ShaderConstant .
 SetValueTranspose	Overloaded. Sets the value of a ShaderConstant to an array.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also








Reference

[ShaderConstant Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstant Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetValue	Overloaded. Sets the value of a ShaderConstant .
	SetValueTranspose	Overloaded. Sets the value of a ShaderConstant to an array.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ShaderConstant Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstant.SetValue Method

Sets the value of a [ShaderConstant](#).

Overload List

Name	Description
ShaderConstant.SetValue (GraphicsDevice, Boolean)	Sets the value of a ShaderConstant to a Boolean .
ShaderConstant.SetValue (GraphicsDevice, Boolean[])	Sets the value of a ShaderConstant to an array of Boolean objects.
ShaderConstant.SetValue (GraphicsDevice, Int32)	Sets the value of a ShaderConstant to an Int32 .
ShaderConstant.SetValue (GraphicsDevice, Int32[])	Sets the value of a ShaderConstant to an array of Int32 objects.
ShaderConstant.SetValue (GraphicsDevice, Matrix)	Sets the value of a ShaderConstant to a Matrix .
ShaderConstant.SetValue (GraphicsDevice, Matrix[])	Sets the value of a ShaderConstant to an array of Matrix objects.
ShaderConstant.SetValue (GraphicsDevice, Quaternion)	Sets the value of a ShaderConstant to a Quaternion .
ShaderConstant.SetValue (GraphicsDevice, Quaternion[])	Sets the value of a ShaderConstant to an array of Quaternion objects.
ShaderConstant.SetValue (GraphicsDevice, Single)	Sets the value of a ShaderConstant to a Single .
ShaderConstant.SetValue (GraphicsDevice, Single[])	Sets the value of a ShaderConstant to an array of Single objects.
ShaderConstant.SetValue (GraphicsDevice, Vector2)	Sets the value of a ShaderConstant to a Vector2 .
ShaderConstant.SetValue (GraphicsDevice, Vector2[])	Sets the value of a ShaderConstant to an array of Vector2 objects.
ShaderConstant.SetValue (GraphicsDevice, Vector3)	Sets the value of a ShaderConstant to a Vector3 .
ShaderConstant.SetValue (GraphicsDevice, Vector3[])	Sets the value of a ShaderConstant to an array of Vector3 objects.
ShaderConstant.SetValue (GraphicsDevice, Vector4)	Sets the value of a ShaderConstant to a Vector4 .
ShaderConstant.SetValue (GraphicsDevice, Vector4[])	Sets the value of a ShaderConstant to an array of Vector4 objects.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstant.SetValue Method (GraphicsDevice, Boolean)

Sets the value of a [ShaderConstant](#) to a [Boolean](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    bool value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

[[MarshalAsAttribute\(U1\)](#)] Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Boolean[])

Sets the value of a [ShaderConstant](#) to an array of [Boolean](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    bool[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Int32)

Sets the value of a [ShaderConstant](#) to an [Int32](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    int value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Int32[])

Sets the value of a [ShaderConstant](#) to an array of [Int32](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    int[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Matrix)

Sets the value of a [ShaderConstant](#) to a [Matrix](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Matrix value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Matrix[])

Sets the value of a [ShaderConstant](#) to an array of [Matrix](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Matrix[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Quaternion)

Sets the value of a [ShaderConstant](#) to a [Quaternion](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Quaternion value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Quaternion[])

Sets the value of a [ShaderConstant](#) to an array of [Quaternion](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Quaternion[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Single)

Sets the value of a [ShaderConstant](#) to a [Single](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    float value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Single[])

Sets the value of a [ShaderConstant](#) to an array of [Single](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    float[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Vector2)

Sets the value of a [ShaderConstant](#) to a [Vector2](#)

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Vector2 value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Vector2[])

Sets the value of a [ShaderConstant](#) to an array of [Vector2](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Vector2[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Vector3)

Sets the value of a [ShaderConstant](#) to a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Vector3 value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Vector3[])

Sets the value of a [ShaderConstant](#) to an array of [Vector3](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Vector3[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Vector4)

Sets the value of a [ShaderConstant](#) to a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Vector4 value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValue Method (GraphicsDevice, Vector4[])

Sets the value of a [ShaderConstant](#) to an array of [Vector4](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValue (  
    GraphicsDevice graphicsDevice,  
    Vector4[] value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

Value to assign to the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValueTranspose Method

Sets the value of a [ShaderConstant](#) to an array.

Overload List

Name	Description
ShaderConstant.SetValueTranspose (GraphicsDevice, Matrix)	Sets the value of a ShaderConstant to a matrix.
ShaderConstant.SetValueTranspose (GraphicsDevice, Matrix[])	Sets the value of a ShaderConstant to an array of matrices.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstant.SetValueTranspose Method (GraphicsDevice, Matrix)

Sets the value of a [ShaderConstant](#) to an matrix.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValueTranspose (  
    GraphicsDevice graphicsDevice,  
    Matrix value  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

The matrix to set the [ShaderConstant](#) to.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SetValueTranspose Method (GraphicsDevice, Matrix[])

Sets the value of a [ShaderConstant](#) to an array of matrices.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetValueTranspose (
    GraphicsDevice graphicsDevice,
    Matrix[] value
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the [ShaderConstant](#).

value

The array of values to set the [ShaderConstant](#) to.

See Also

Reference

[ShaderConstant Class](#)












[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant Properties

Public Properties

	Name	Description
	ColumnCount	Gets the number of columns in the ShaderConstant .
	ElementCount	Gets the number of elements in the ShaderConstant .
	Name	Gets the name of the ShaderConstant .
	ParameterClass	Gets the EffectParameterClass of the ShaderConstant .
	ParameterType	Gets the EffectParameterType of the ShaderConstant .
	RegisterCount	Gets the number of registers that contain data.
	RegisterIndex	Gets the index of the ShaderConstant in the constant table.
	RegisterSet	Gets the data type of the shader register.
	RowCount	Gets the number of rows in the ShaderConstant .
	SamplerIndex	Gets the sampler index number of the ShaderConstant from the constant table.
	StructureMemberCount	Gets the number of structure member subparameters.

See Also

Reference

[ShaderConstant Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstant.ColumnCount Property

Gets the number of columns in the [ShaderConstant](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ColumnCount { get; }
```

Property Value

The number of columns in the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.ElementCount Property

Gets the number of elements in the [ShaderConstant](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ElementCount { get; }
```

Property Value

The number of elements in the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.Name Property

Gets the name of the [ShaderConstant](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.ParameterClass Property

Gets the [EffectParameterClass](#) of the [ShaderConstant](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterClass ParameterClass { get; }
```

Property Value

The [EffectParameterClass](#) of the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.ParameterType Property

Gets the [EffectParameterType](#) of the [ShaderConstant](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public EffectParameterType ParameterType { get; }
```

Property Value

The [EffectParameterType](#) of the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.RegisterCount Property

Gets the number of registers that contain data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int RegisterCount { get; }
```

Property Value

The number of registers that contain data.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.RegisterIndex Property

Gets the index of the [ShaderConstant](#) in the constant table.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int RegisterIndex { get; }
```

Property Value

The index of the [ShaderConstant](#) in the constant table.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.RegisterSet Property

Gets the data type of the shader register.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderRegisterSet RegisterSet { get; }
```

Property Value

The data type of the shader register.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.RowCount Property

Gets the number of rows in the [ShaderConstant](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int RowCount { get; }
```

Property Value

The number of rows in the [ShaderConstant](#).

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.SamplerIndex Property

Gets the sampler index number of the [ShaderConstant](#) from the constant table.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int SamplerIndex { get; }
```

Property Value

The sampler index number of the [ShaderConstant](#) from the constant table.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstant.StructureMemberCount Property

Gets the number of structure member subparameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int StructureMemberCount { get; }
```

Property Value

The number of structure member subparameters.

See Also

Reference

[ShaderConstant Class](#)

[ShaderConstant Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantCollection Class

Manipulates a collection of [ShaderConstant](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ShaderConstantCollection : IEnumerable<ShaderConstant>
```

See Also

Reference

[ShaderConstantCollection Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista







ShaderConstantCollection Members

The following tables list the members exposed by the ShaderConstantCollection type.



Public Properties

	Name	Description
	Count	Gets the number of ShaderConstant objects in this ShaderConstantCollection .
	Item	Overloaded. Gets a specific ShaderConstant object.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the ShaderConstantCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[ShaderConstantCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that can iterate through the ShaderConstantCollection .
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ShaderConstantCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantCollection.GetEnumerator Method

Returns an enumerator that can iterate through the [ShaderConstantCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<ShaderConstant> GetEnumerator ()
```

Return Value

The iterator.

See Also

Reference

[ShaderConstantCollection Class](#)



[ShaderConstantCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of ShaderConstant objects in this ShaderConstantCollection .
	Item	Overloaded. Gets a specific ShaderConstant object.

See Also

Reference

[ShaderConstantCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantCollection.Count Property

Gets the number of [ShaderConstant](#) objects in this [ShaderConstantCollection](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of [ShaderConstant](#) objects in this [ShaderConstantCollection](#).

See Also

Reference

[ShaderConstantCollection Class](#)

[ShaderConstantCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantCollection.Item Property

Gets a specific [ShaderConstant](#) object.

Overload List

Name	Description
ShaderConstantCollection.Item (Int32)	Gets a specific ShaderConstant object using an index value.
ShaderConstantCollection.Item (String)	Gets a specific ShaderConstant object using an index value.

See Also

Reference

[ShaderConstantCollection Class](#)

[ShaderConstantCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantCollection.Item Property (Int32)

Gets a specific [ShaderConstant](#) object using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderConstant this [
    int index
] { get; }
```

Property Value

The [ShaderConstant](#) object at the specified index.

See Also

Reference

[ShaderConstantCollection Class](#)

[ShaderConstantCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantCollection.Item Property (String)

Gets a specific [ShaderConstant](#) object using an index value.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderConstant this [
    string name
] { get; }
```

Property Value

The [ShaderConstant](#) object specified by **name**.

See Also

Reference

[ShaderConstantCollection Class](#)

[ShaderConstantCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable Class

Contains the variables that are used by high-level language shaders and effects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class ShaderConstantTable : IDisposable
```

See Also

Reference

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





ShaderConstantTable Members

The following tables list the members exposed by the ShaderConstantTable type.








Public Constructors

Name	Description
 ShaderConstantTable	Initializes a new instance of the ShaderConstantTable class.




Public Properties

Name	Description
 Constants	Gets a collection of the constants in the ShaderConstantTable .
 Creator	Gets the name of the constant table creator.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Version	Gets the shader version.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetDefaults	Sets the constants in the ShaderConstantTable to their default values. The default values are declared in the variable declarations in the shader.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[ShaderConstantTable Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantTable Constructor

Initializes a new instance of the [ShaderConstantTable](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderConstantTable (  
    byte[] code  
)
```

Parameters

code

The shader byte code.

Exceptions

Exception type	Condition
ArgumentException	The array <i>code</i> must have a length that is a multiple of four.
ArgumentNullException	<i>code</i> is null .
InvalidOperationException	Unable to create the ShaderConstantTable .

See Also

Reference

[ShaderConstantTable Class](#)








[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetDefaults	Sets the constants in the ShaderConstantTable to their default values. The default values are declared in the variable declarations in the shader.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[ShaderConstantTable Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantTable.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
ShaderConstantTable.Dispose ()	Immediately releases the unmanaged resources used by this object.
ShaderConstantTable.Dispose (Boolean)	Releases the unmanaged resources used by the ShaderConstantTable and optionally releases the managed resources.

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantTable.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable.Dispose Method (Boolean)

Releases the unmanaged resources used by the [ShaderConstantTable](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\] true](#) to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [ShaderConstantTable](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderConstantTable.SetDefaults Method

Sets the constants in the [ShaderConstantTable](#) to their default values. The default values are declared in the variable declarations in the shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetDefaults (  
    GraphicsDevice device  
)
```

Parameters

device

The [GraphicsDevice](#) associated with the [ShaderConstantTable](#).

See Also

Reference

[ShaderConstantTable Class](#)





[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable Properties

Public Properties

	Name	Description
	Constants	Gets a collection of the constants in the ShaderConstantTable .
	Creator	Gets the name of the constant table creator.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Version	Gets the shader version.

See Also

Reference

[ShaderConstantTable Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantTable.Constants Property

Gets a collection of the constants in the [ShaderConstantTable](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ShaderConstantCollection Constants { get; }
```

Property Value

The collection of constants in the [ShaderConstantTable](#).

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable.Creator Property

Gets the name of the constant table creator.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Creator { get; }
```

Property Value

The name of the constant table creator.

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable.Version Property

Gets the shader version.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Version Version { get; }
```

Property Value

The shader version.

See Also

Reference

[ShaderConstantTable Class](#)


[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderConstantTable Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[ShaderConstantTable Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderConstantTable.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[ShaderConstantTable Class](#)

[ShaderConstantTable Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderProfile Enumeration

Defines vertex and pixel shader versions.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ShaderProfile
```

Members

Member name	Description
PS_1_1	Pixel shader version ps_1_1.
PS_1_2	Pixel shader version ps_1_2.
PS_1_3	Pixel shader version ps_1_3.
PS_1_4	Pixel shader version ps_1_4.
PS_2_0	Pixel shader version ps_2_0.
PS_2_A	Pixel shader version ps_2_a.
PS_2_B	Pixel shader version ps_2_b.
PS_2_SW	Pixel software shader version ps_2_sw.
PS_3_0	Pixel shader version ps_3_0.
Unknown	Unknown pixel shader version.
VS_1_1	Vertex shader version v_1_1.
VS_2_0	Vertex shader version v_2_0.
VS_2_A	Vertex shader version v_2_a.
VS_2_SW	Vertex software shader version v_2_sw.
VS_3_0	Vertex shader version vs_3_0.
XPS_3_0	Xbox microcode assembly pixel shader version xps_3_0. Microcode assembly language supports a superset of the ps_3_0 and vs_3_0 specifications defined by Direct3D 9.0 for Windows. It does not support earlier Direct3D vertex and pixel shader specifications.
XVS_3_0	Xbox microcode assembly vertex shader version xvs_3_0. Microcode assembly language supports a superset of the ps_3_0 and vs_3_0 specifications defined by Direct3D 9.0 for Windows. It does not support earlier Direct3D vertex and pixel shader specifications.

Remarks

These pixel shader versions identify the assembly-level shader model compilation target the HLSL compiler should use to express the final shader code. When shader compilation is completed at runtime, an application might query the using [MaxVertexShaderProfile](#) or [MaxPixelShaderProfile](#) to select a compilation target to match the capabilities of the graphics device.

See Also

Tasks

[How To: Check for Shader Model 2.0 Support](#)

Reference

[GraphicsDeviceCapabilities.MaxPixelShaderProfile](#) Property

[GraphicsDeviceCapabilities.MaxVertexShaderProfile](#) Property

[ShaderCompiler.CompileFromFile](#) Method

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ShaderRegisterSet Enumeration

Defines the data type of a shader register.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ShaderRegisterSet
```

Members

Member name	Description
Boolean	Boolean value.
Float4	4D floating-point number.
Int4	4D integer number.
Sampler	The register contains 4D sampler data.

See Also

Reference

[ShaderConstant.RegisterSet Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

ShaderSemantic Structure

Note

This structure is available only when developing for Windows.

Semantics map a parameter to vertex or pixel shader registers. They can also be optional descriptive strings attached to non-register parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct ShaderSemantic
```

RemarksSemantics are required for vertex and pixel shader, input and output registers.

See Also

Reference

[ShaderSemantic Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista





ShaderSemantic Members

The following tables list the members exposed by the ShaderSemantic type.



Public Properties

	Name	Description
	UsageIndex	Gets or sets options that modify how the usage is interpreted. The usage and usage index make up a vertex declaration.
	VertexElementUsage	Gets or sets options that identify how resources are used.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[ShaderSemantic Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderSemantic Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ShaderSemantic Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderSemantic.ToString Method

Note

This method is available only when developing for Windows.

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[ShaderSemantic Structure](#)



[ShaderSemantic Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderSemantic Properties

Public Properties

	Name	Description
	UsageIndex	Gets or sets options that modify how the usage is interpreted. The usage and usage index make up a vertex declaration.
	VertexElementUsage	Gets or sets options that identify how resources are used.

See Also

Reference

[ShaderSemantic Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

ShaderSemantic.UsageIndex Property

Note

This property is available only when developing for Windows.

Gets or sets options that modify how the usage is interpreted. The usage and usage index make up a vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int UsageIndex { get; set; }
```

Property Value

Options that modify how the usage is interpreted. The usage and usage index make up a vertex declaration.

See Also

Reference

[ShaderSemantic Structure](#)

[ShaderSemantic Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ShaderSemantic.VertexElementUsage Property

Note

This property is available only when developing for Windows.

Gets or sets options that identify how resources are used.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexElementUsage VertexElementUsage { get; set; }
```

Property Value

Options that identify how resources are used.

See Also

Reference

[ShaderSemantic Structure](#)

[ShaderSemantic Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SpriteBatch Class

Enables a group of sprites to be drawn using the same settings.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class SpriteBatch : IDisposable
```

See Also

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





SpriteBatch Members

The following tables list the members exposed by the SpriteBatch type.











Public Constructors

Name	Description
 SpriteBatch	Initializes a new instance of the class, which enables a group of sprites to be drawn using the same settings.



Public Properties

Name	Description
 GraphicsDevice	Gets the graphics device associated with this SpriteBatch .
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Name	Gets or sets the name of this sprite batch.
 Tag	Gets or sets an object that uniquely identifies this sprite batch.


Public Methods

Name	Description
 Begin	Overloaded. Prepares the graphics device for drawing sprites.
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Draw	Overloaded. Adds a sprite to the batch of sprites to be rendered.
 DrawString	Overloaded. Adds a sprite string to the batch of sprites to be rendered.
 End	Flushes the sprite batch and restores the device state to how it was before Begin was called.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch Constructor

Initializes a new instance of the class, which enables a group of sprites to be drawn using the same settings.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SpriteBatch (  
    GraphicsDevice graphicsDevice  
)
```

Parameters

graphicsDevice

The graphics device where sprites will be drawn.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)











[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch Methods

Public Methods

	Name	Description
	Begin	Overloaded. Prepares the graphics device for drawing sprites.
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Draw	Overloaded. Adds a sprite to the batch of sprites to be rendered.
	DrawString	Overloaded. Adds a sprite string to the batch of sprites to be rendered.
	End	Flushes the sprite batch and restores the device state to how it was before Begin was called.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SpriteBatch Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.Begin Method

Prepares the graphics device for drawing sprites.

Unexpected behavior can result when rendering sprites and 3D objects on the same graphics device. For more information, see [Render States and the Graphics Device](#).

Overload List

Name	Description
SpriteBatch.Begin ()	Prepares the graphics device for drawing sprites.
SpriteBatch.Begin (SpriteBlendMode)	Prepares the graphics device for drawing sprites with specified blending options.
SpriteBatch.Begin (SpriteBlendMode, SpriteSortMode, SaveStateMode)	Prepares the graphics device for drawing sprites with specified blending, sorting, and render state options.
SpriteBatch.Begin (SpriteBlendMode, SpriteSortMode, SaveStateMode, Matrix)	Prepares the graphics device for drawing sprites with specified blending, sorting, and render state options, and a global transform matrix.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.Begin Method ()

Prepares the graphics device for drawing sprites.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin ()
```

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

Remarks

This method must be called before any calls to [Draw](#). When all the sprites have been drawn, call [End](#).

Begin sets the following render states:

- [CullMode](#)
- [DepthBufferEnable](#)
- [AlphaBlendEnable](#)
- [AlphaTestEnable](#)
- [AlphaBlendOperation](#)
- [SourceBlend](#)
- [DestinationBlend](#)
- [SeparateAlphaBlendEnabled](#)
- [AlphaFunction](#)
- [ReferenceAlpha](#)

Begin sets the following sampler states:

- [AddressU](#)
- [AddressV](#)
- [MagFilter](#)
- [MinFilter](#)
- [MipFilter](#)
- [MipMapLevelOfDetailBias](#)
- [MaxMipLevel](#)

This version of **Begin** sets [SpriteBlendMode](#) to **AlphaBlend**, [SpriteSortMode](#) to **Deferred**, and [SaveStateMode](#) to **None**. When [SaveStateMode](#) is set to **None**, the caller must reset the render and sampler states if they are used elsewhere.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Begin Method (SpriteBlendMode)

Prepares the graphics device for drawing sprites with specified blending options.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin (
    SpriteBlendMode blendMode
)
```

Parameters

blendMode

Blending options to use when rendering.

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

Remarks

This method must be called before any calls to [Draw](#). When all the sprites have been drawn, call [End](#).

Begin sets the following render states:

- [CullMode](#)
- [DepthBufferEnable](#)
- [AlphaBlendEnable](#)
- [AlphaTestEnable](#)
- [AlphaBlendOperation](#)
- [SourceBlend](#)
- [DestinationBlend](#)
- [SeparateAlphaBlendEnabled](#)
- [AlphaFunction](#)
- [ReferenceAlpha](#)

Begin sets the following sampler states:

- [AddressU](#)
- [AddressV](#)
- [MagFilter](#)
- [MinFilter](#)
- [MipFilter](#)
- [MipMapLevelOfDetailBias](#)
- [MaxMipLevel](#)

This version of **Begin** sets [SpriteSortMode](#) to **Deferred**, and [SaveStateMode](#) to **None**. When [SaveStateMode](#) is set to **None**, the caller must reset the render and sampler states if they are used elsewhere.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Begin Method (SpriteBlendMode, SpriteSortMode, SaveStateMode)

Prepares the graphics device for drawing sprites with specified blending, sorting, and render state options.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin (
    SpriteBlendMode blendMode,
    SpriteSortMode sortMode,
    SaveStateMode stateMode
)
```

Parameters

blendMode

Blending options to use when rendering.

sortMode

Sorting options to use when rendering.

stateMode

Rendering state options.

Exceptions

Exception type	Condition
InvalidOperationException	Begin has been called before calling End after the last call to Begin . Begin cannot be called again until End has been successfully called.

Remarks

This method must be called before any calls to [Draw](#). When all the sprites have been drawn, call [End](#).

Begin sets the following render states:

- [CullMode](#)
- [DepthBufferEnable](#)
- [AlphaBlendEnable](#)
- [AlphaTestEnable](#)
- [AlphaBlendOperation](#)
- [SourceBlend](#)
- [DestinationBlend](#)
- [SeparateAlphaBlendEnabled](#)
- [AlphaFunction](#)
- [ReferenceAlpha](#)

Begin sets the following sampler states:

- [AddressU](#)
- [AddressV](#)
- [MagFilter](#)
- [MinFilter](#)
- [MipFilter](#)
- [MipMapLevelOfDetailBias](#)
- [MaxMipLevel](#)

If **Begin** is called with [SaveStateMode](#) set to **None**, the caller must reset these states must be reset if they are used elsewhere. Calling **Begin** with [SaveStateMode](#) set to **SaveState** will reset these values properly, but performance may be adversely affected.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Begin Method (SpriteBlendMode, SpriteSortMode, SaveStateMode, Matrix)

Prepares the graphics device for drawing sprites with specified blending, sorting, and render state options, and a global transform matrix.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Begin (  
    SpriteBlendMode blendMode,  
    SpriteSortMode sortMode,  
    SaveStateMode stateMode,  
    Matrix transformMatrix  
)
```

Parameters

blendMode

Blending options to use when rendering.

sortMode

Sorting options to use when rendering.

stateMode

Rendering state options.

transformMatrix

A matrix to apply to position, rotation, scale, and depth data passed to [Draw](#).

Remarks

This method must be called before any calls to [Draw](#). When all the sprites have been drawn, call [End](#).

Begin sets the following render states.

- [CullMode](#)
- [DepthBufferEnable](#)
- [AlphaBlendEnable](#)
- [AlphaTestEnable](#)
- [AlphaBlendOperation](#)
- [SourceBlend](#)
- [DestinationBlend](#)
- [SeparateAlphaBlendEnabled](#)
- [AlphaFunction](#)
- [ReferenceAlpha](#)

Begin sets the following sampler states.

- [AddressU](#)
- [AddressV](#)
- [MagFilter](#)
- [MinFilter](#)
- [MipFilter](#)
- [MipMapLevelOfDetailBias](#)
- [MaxMipLevel](#)

If **Begin** is called with [SaveStateMode](#) set to **None**, then these states must be reset by the caller if they are used elsewhere. Calling **Begin** with [SaveStateMode](#) set to **SaveState** will reset these values properly, but may adversely affect performance.

Use **transformMatrix** to apply the same transformation to all sprites drawn by this [SpriteBatch](#). This matrix can be any combination of translation, scaling, or rotation operations. This matrix is applied to the position, rotation, scale, and depth

values provided to [Draw](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
SpriteBatch.Dispose ()	Immediately releases the unmanaged resources used by this object.
SpriteBatch.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method

Adds a sprite to the batch of sprites to be rendered.

Overload List

Name	Description
SpriteBatch.Draw (Texture2D, Rectangle, Color)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, destination rectangle, and color tint.
SpriteBatch.Draw (Texture2D, Rectangle, Nullable<Rectangle>, Color)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, destination and source rectangles, and color tint.
SpriteBatch.Draw (Texture2D, Rectangle, Nullable<Rectangle>, Color, Single, Vector2, SpriteEffects, Single)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, destination, and source rectangles, color tint, rotation, origin, effects, and sort depth.
SpriteBatch.Draw (Texture2D, Vector2, Color)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, and color tint.
SpriteBatch.Draw (Texture2D, Vector2, Nullable<Rectangle>, Color)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, source rectangle, and color tint.
SpriteBatch.Draw (Texture2D, Vector2, Nullable<Rectangle>, Color, Single, Vector2, Single, SpriteEffects, Single)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, optional source rectangle, color tint, rotation, origin, scale, effects, and sort depth.
SpriteBatch.Draw (Texture2D, Vector2, Nullable<Rectangle>, Color, Single, Vector2, Vector2, SpriteEffects, Single)	Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, source rectangle, color tint, rotation, origin, scale, effects, and sort depth.

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.Draw Method (Texture2D, Rectangle, Color)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, destination rectangle, and color tint.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (  
    Texture2D texture,  
    Rectangle destinationRectangle,  
    Color color  
)
```

Parameters

texture

The sprite texture.

destinationRectangle

A rectangle specifying, in screen coordinates, where the sprite will be drawn. If this rectangle is not the same size as *sourceRectangle*, the sprite is scaled to fit.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before Draw can be called.

Remarks

Before any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method (Texture2D, Rectangle, Nullable<Rectangle>, Color)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, destination and source rectangles, and color tint.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (
    Texture2D texture,
    Rectangle destinationRectangle,
    Nullable<Rectangle> sourceRectangle,
    Color color
)
```

Parameters

texture

The sprite texture.

destinationRectangle

A rectangle specifying, in screen coordinates, where the sprite will be drawn. If this rectangle is not the same size as *sourceRectangle* the sprite will be scaled to fit.

sourceRectangle

A rectangle specifying, in texels, which section of the rectangle to draw. Use **null** to draw the entire texture.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before Draw can be called.

Remarks

Before any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method (Texture2D, Rectangle, Nullable<Rectangle>, Color, Single, Vector2, SpriteEffects, Single)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, destination, and source rectangles, color tint, rotation, origin, effects, and sort depth.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (
    Texture2D texture,
    Rectangle destinationRectangle,
    Nullable<Rectangle> sourceRectangle,
    Color color,
    float rotation,
    Vector2 origin,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

texture

The sprite texture.

destinationRectangle

A rectangle specifying, in screen coordinates, where the sprite will be drawn. If this rectangle is not the same size as *sourceRectangle*, the sprite is scaled to fit.

sourceRectangle

A rectangle specifying, in texels, which section of the rectangle to draw. Use **null** to draw the entire texture.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

rotation

The angle, in radians, to rotate the sprite around the origin.

origin

The origin of the sprite. Specify (0,0) for the upper-left corner.

effects

Rotations to apply prior to rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

You must specify either [SpriteSortMode.FrontToBack](#) or [SpriteSortMode.BackToFront](#) for this parameter to affect sprite drawing.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before you can call Draw .

Remarks

Before making any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method (Texture2D, Vector2, Color)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, and color tint.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (  
    Texture2D texture,  
    Vector2 position,  
    Color color  
)
```

Parameters

texture

The sprite texture.

position

The location, in screen coordinates, where the sprite will be drawn.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before Draw can be called.

Remarks

Before any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method (Texture2D, Vector2, Nullable<Rectangle>, Color)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, source rectangle, and color tint.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (  
    Texture2D texture,  
    Vector2 position,  
    Nullable<Rectangle> sourceRectangle,  
    Color color  
)
```

Parameters

texture

The sprite texture.

position

The location, in screen coordinates, where the sprite will be drawn.

sourceRectangle

A rectangle specifying, in texels, which section of the rectangle to draw. Use **null** to draw the entire texture.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before Draw can be called.

Remarks

Before any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method (Texture2D, Vector2, Nullable<Rectangle>, Color, Single, Vector2, Single, SpriteEffects, Single)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, optional source rectangle, color tint, rotation, origin, scale, effects, and sort depth.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (
    Texture2D texture,
    Vector2 position,
    Nullable<Rectangle> sourceRectangle,
    Color color,
    float rotation,
    Vector2 origin,
    float scale,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

texture

The sprite texture.

position

The location, in screen coordinates, where the sprite will be drawn.

sourceRectangle

A rectangle specifying, in texels, which section of the rectangle to draw. Use **null** to draw the entire texture.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

rotation

The angle, in radians, to rotate the sprite around the origin.

origin

The origin of the sprite. Specify (0,0) for the upper-left corner.

scale

Uniform multiple by which to scale the sprite width and height.

effects

Rotations to apply prior to rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

You must specify either [SpriteSortMode.FrontToBack](#) or [SpriteSortMode.BackToFront](#) for this parameter to affect sprite drawing.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before you can call Draw .

Remarks

Before making any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Draw Method (Texture2D, Vector2, Nullable<Rectangle>, Color, Single, Vector2, Vector2, SpriteEffects, Single)

Adds a sprite to the batch of sprites to be rendered, specifying the texture, screen position, source rectangle, color tint, rotation, origin, scale, effects, and sort depth.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Draw (
    Texture2D texture,
    Vector2 position,
    Nullable<Rectangle> sourceRectangle,
    Color color,
    float rotation,
    Vector2 origin,
    Vector2 scale,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

texture

The sprite texture.

position

The location, in screen coordinates, where the sprite will be drawn.

sourceRectangle

A rectangle specifying, in texels, which section of the rectangle to draw. Use **null** to draw the entire texture.

color

The color channel modulation to use. Use [Color.White](#) for full color with no tinting.

rotation

The angle, in radians, to rotate the sprite around the origin.

origin

The origin of the sprite. Specify (0,0) for the upper-left corner.

scale

Vector containing separate scalar multiples for the x- and y-axes of the sprite.

effects

Rotations to apply before rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

You must specify either [SpriteSortMode.FrontToBack](#) or [SpriteSortMode.BackToFront](#) for this parameter to affect sprite drawing.

Exceptions

Exception type	Condition
ArgumentNullException	<i>texture</i> is null .
InvalidOperationException	Draw was called, but Begin has not yet been called. Begin must be called successfully before you can call Draw .

Remarks

Before making any calls to [Draw](#), you must call [Begin](#). Once all calls to [Draw](#) are complete, call [End](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.DrawString Method

Adds a sprite string to the batch of sprites to be rendered.

Overload List

Name	Description
SpriteBatch.DrawString (SpriteFont, String, Vector2, Color)	Adds a sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, and color tint.
SpriteBatch.DrawString (SpriteFont, String, Vector2, Color, Single, Vector2, Single, SpriteEffects, Single)	Adds a sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, and effects.
SpriteBatch.DrawString (SpriteFont, String, Vector2, Color, Single, Vector2, Vector2, SpriteEffects, Single)	Adds a sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, effects, and depth.
SpriteBatch.DrawString (SpriteFont, StringBuilder, Vector2, Color)	Adds a mutable sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, and color tint.
SpriteBatch.DrawString (SpriteFont, StringBuilder, Vector2, Color, Single, Vector2, Single, SpriteEffects, Single)	Adds a mutable sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, and effects.
SpriteBatch.DrawString (SpriteFont, StringBuilder, Vector2, Color, Single, Vector2, Vector2, SpriteEffects, Single)	Adds a mutable sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, effects, and depth.

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.DrawString Method (SpriteFont, String, Vector2, Color)

Adds a sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, and color tint.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawString (
    SpriteFont spriteFont,
    string text,
    Vector2 position,
    Color color
)
```

Parameters

spriteFont

The sprite font.

text

The string to draw.

position

The location, in screen coordinates, where the text will be drawn.

color

The desired color of the text.

Exceptions

Exception type	Condition
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .
ArgumentNullException	<i>spriteFont</i> or <i>text</i> is null .
InvalidOperationException	DrawString was called, but Begin has not yet been called. Begin must be called successfully before DrawString can be called.

Remarks

[SpriteFont](#) objects are loaded from the Content Manager. See the [SpriteFont](#) class for more information.

Before any calls to [DrawString](#), you must call [Begin](#). Once all calls to [DrawString](#) are complete, call [End](#).

Use a newline character (\n) to draw more than one line of text.

See Also

Tasks

[How To: Draw Text](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.DrawString Method (SpriteFont, String, Vector2, Color, Single, Vector2, Single, SpriteEffects, Single)

Adds a sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, and effects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawString (
    SpriteFont spriteFont,
    string text,
    Vector2 position,
    Color color,
    float rotation,
    Vector2 origin,
    float scale,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

spriteFont

The sprite font.

text

The string to draw.

position

The location, in screen coordinates, where the text will be drawn.

color

The desired color of the text.

rotation

The angle, in radians, to rotate the text around the origin.

origin

The origin of the string. Specify (0,0) for the upper-left corner.

scale

Uniform multiple by which to scale the sprite width and height.

effects

Rotations to apply prior to rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

Exceptions

Exception type	Condition
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .
ArgumentNullException	<i>spriteFont</i> or <i>text</i> is null .
InvalidOperationException	DrawString was called, but Begin has not yet been called. Begin must be called successfully before DrawString can be called.

Remarks

[SpriteFont](#) objects are loaded from the Content Manager. See the [SpriteFont](#) class for more information.

Before any calls to [DrawString](#), you must call [Begin](#). Once all calls to [DrawString](#) are complete, call [End](#).

Use a newline character (\n) to draw more than one line of text.

See Also

Tasks

[How To: Draw Text](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.DrawString Method (SpriteFont, String, Vector2, Color, Single, Vector2, Vector2, SpriteEffects, Single)

Adds a sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, effects, and depth.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawString (
    SpriteFont spriteFont,
    string text,
    Vector2 position,
    Color color,
    float rotation,
    Vector2 origin,
    Vector2 scale,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

spriteFont

The sprite font.

text

The string to draw.

position

The location, in screen coordinates, where the text will be drawn.

color

The desired color of the text.

rotation

The angle, in radians, to rotate the text around the origin.

origin

The origin of the string. Specify (0,0) for the upper-left corner.

scale

Vector containing separate scalar multiples for the x- and y-axes of the sprite.

effects

Rotations to apply prior to rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

Exceptions

Exception type	Condition
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .
ArgumentNullException	<i>spriteFont</i> or <i>text</i> is null .
InvalidOperationException	DrawString was called, but Begin has not yet been called. Begin must be called successfully before DrawString can be called.

Remarks

[SpriteFont](#) objects are loaded from the Content Manager. See the [SpriteFont](#) class for more information.

Before any calls to [DrawString](#), you must call [Begin](#). Once all calls to [DrawString](#) are complete, call [End](#).

Use a newline character (\n) to draw more than one line of text.

See Also

Tasks

[How To: Draw Text](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.DrawString Method (SpriteFont, StringBuilder, Vector2, Color)

Adds a mutable sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, and color tint.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawString (
    SpriteFont spriteFont,
    StringBuilder text,
    Vector2 position,
    Color color
)
```

Parameters

spriteFont

The sprite font.

text

The mutable (read/write) string to draw.

position

The location, in screen coordinates, where the text will be drawn.

color

The desired color of the text.

Exceptions

Exception type	Condition
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .
ArgumentNullException	<i>spriteFont</i> or <i>text</i> is null .
InvalidOperationException	DrawString was called, but Begin has not yet been called. Begin must be called successfully before DrawString can be called.

Remarks

By accepting a [StringBuilder](#) argument in place of a [String](#) argument, this form supports measurement of dynamically formatted text.

[SpriteFont](#) objects are loaded from the Content Manager. See the [SpriteFont](#) class for more information.

Before any calls to [DrawString](#), you must call [Begin](#). Once all calls to [DrawString](#) are complete, call [End](#).

Use a newline character (\n) to draw more than one line of text.

See Also

Tasks

[How To: Draw Text](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.DrawString Method (SpriteFont, StringBuilder, Vector2, Color, Single, Vector2, Single, SpriteEffects, Single)

Adds a mutable sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, and effects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawString (
    SpriteFont spriteFont,
    StringBuilder text,
    Vector2 position,
    Color color,
    float rotation,
    Vector2 origin,
    float scale,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

spriteFont

The sprite font.

text

The mutable (read/write) string to draw.

position

The location, in screen coordinates, where the text will be drawn.

color

The desired color of the text.

rotation

The angle, in radians, to rotate the text around the origin.

origin

The origin of the string. Specify (0,0) for the upper-left corner.

scale

Uniform multiple by which to scale the sprite width and height.

effects

Rotations to apply prior to rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

Exceptions

Exception type	Condition
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .
ArgumentNullException	<i>spriteFont</i> or <i>text</i> is null .
InvalidOperationException	DrawString was called, but Begin has not yet been called. Begin must be called successfully before DrawString can be called.

Remarks

By accepting a [StringBuilder](#) argument in place of a [String](#) argument, this form supports measurement of dynamically formatted text.

[SpriteFont](#) objects are loaded from the Content Manager. See the [SpriteFont](#) class for more information.

Before any calls to [DrawString](#), you must call [Begin](#). Once all calls to [DrawString](#) are complete, call [End](#).

Use a newline character (\n) to draw more than one line of text.

See Also

Tasks

[How To: Draw Text](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.DrawString Method (SpriteFont, StringBuilder, Vector2, Color, Single, Vector2, Vector2, SpriteEffects, Single)

Adds a mutable sprite string to the batch of sprites to be rendered, specifying the font, output text, screen position, color tint, rotation, origin, scale, effects, and depth.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void DrawString (
    SpriteFont spriteFont,
    StringBuilder text,
    Vector2 position,
    Color color,
    float rotation,
    Vector2 origin,
    Vector2 scale,
    SpriteEffects effects,
    float layerDepth
)
```

Parameters

spriteFont

The sprite font.

text

The mutable (read/write) string to draw.

position

The location, in screen coordinates, where the text will be drawn.

color

The desired color of the text.

rotation

The angle, in radians, to rotate the text around the origin.

origin

The origin of the string. Specify (0,0) for the upper-left corner.

scale

Vector containing separate scalar multiples for the x- and y-axes of the sprite.

effects

Rotations to apply prior to rendering.

layerDepth

The sorting depth of the sprite, between 0 (front) and 1 (back).

Exceptions

Exception type	Condition
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .
ArgumentNullException	<i>spriteFont</i> or <i>text</i> is null .
InvalidOperationException	DrawString was called, but Begin has not yet been called. Begin must be called successfully before DrawString can be called.

Remarks

By accepting a [StringBuilder](#) argument in place of a [String](#) argument, this form supports measurement of dynamically formatted text.

[SpriteFont](#) objects are loaded from the Content Manager. See the [SpriteFont](#) class for more information.

Before any calls to [DrawString](#), you must call [Begin](#). Once all calls to [DrawString](#) are complete, call [End](#).

Use a newline character (\n) to draw more than one line of text.

See Also

Tasks

[How To: Draw Text](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.End Method

Flushes the sprite batch and restores the device state to how it was before [Begin](#) was called.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void End ()
```

Exceptions

Exception type	Condition
InvalidOperationException	End was called, but Begin has not yet been called. You must call Begin successfully before you can call End .

Remarks

Call **End** after all calls to [Draw](#) are complete.

Unexpected behavior can result when rendering sprites and 3D objects on the same graphics device. For more information, see [Render States and the Graphics Device](#).

See Also

Tasks

[How To: Draw a Sprite](#)

Concepts

[2D Graphics Overview](#)

Reference

[SpriteBatch Class](#)





[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the graphics device associated with this SpriteBatch .
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Name	Gets or sets the name of this sprite batch.
	Tag	Gets or sets an object that uniquely identifies this sprite batch.

See Also

Reference

[SpriteBatch Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.GraphicsDevice Property

Gets the graphics device associated with this [SpriteBatch](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The graphics device associated with this [SpriteBatch](#).

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Name Property

Gets or sets the name of this sprite batch.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of this sprite batch.

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch.Tag Property

Gets or sets an object that uniquely identifies this sprite batch.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

An object that uniquely identifies this sprite batch.

See Also

Reference

[SpriteBatch Class](#)


[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBatch Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[SpriteBatch Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteBatch.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[SpriteBatch Class](#)

[SpriteBatch Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteBlendMode Enumeration

The following flags are used to specify sprite blending rendering options to the flags parameter in [Begin](#):

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum SpriteBlendMode
```

Members

Member name	Description
Additive	Enable additive blending.
AlphaBlend	Enable alpha blending.
None	No blending specified.

See Also

Tasks

[How To: Draw a Masked Sprite over a Background](#)

Reference

[SpriteBatch.Begin Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteEffects Enumeration

Defines sprite mirroring options.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]  
public enum SpriteEffects
```

Members

Member name	Description
FlipHorizontally	Rotate 180 degrees about the Y axis before rendering.
FlipVertically	Rotate 180 degrees about the X axis before rendering.
None	No rotations specified.

See Also

Reference

[SpriteBatch.Draw Method \(Texture2D, Rectangle, Nullable<Rectangle>, Color, Single, Vector2, SpriteEffects, Single\)](#)

[SpriteBatch.Draw Method \(Texture2D, Vector2, Nullable<Rectangle>, Color, Single, Vector2, Single, SpriteEffects, Single\)](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteFont Class

Represents a font texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SpriteFont
```

Remarks

To load a [SpriteFont](#), click **Sprite Font** on the **Add New Item** menu. This adds an XML file to your project describing how to build a texture map for your font. At build time, XNA Game Studio creates a texture with the image of the characters of the font you specify, with the specified font point size.

At run time, load the font using [ContentManager.Load](#), and pass it to [SpriteBatch.DrawString](#) when drawing text.

Note

As with most types of software, font files are licensed rather than sold. Font licenses vary from vendor to vendor, but most don't allow redistribution of the fonts, and that includes redistribution of reproductions such as bitmaps containing the rasterized character set. This is even true of many of the licenses covering fonts that Microsoft supplies with applications and Windows. Be careful, therefore, to ensure that you have the required license rights to redistribute any font you include as a bitmap containing the rasterized character set in your game!

For an example of rendering a sprite font inside your game, see [How To: Draw Text](#).

See Also

Tasks

[How To: Draw Text](#)

Reference

[SpriteFont Members](#)

[ContentManager.Load](#)

[SpriteBatch.DrawString](#)





[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







SpriteFont Members

The following tables list the members exposed by the SpriteFont type.



Public Properties

	Name	Description
	Characters	Gets a collection of all the characters that are included in the font.
	DefaultCharacter	Gets or sets the default character for the font.
	LineSpacing	The line spacing, in pixels, of this font.
	Spacing	Gets or sets the spacing of the font characters.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MeasureString	Overloaded. Returns the height and width of a given string.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[SpriteFont Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteFont Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MeasureString	Overloaded. Returns the height and width of a given string.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SpriteFont Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteFont.MeasureString Method

Returns the height and width of a given string.

Overload List

Name	Description
SpriteFont.MeasureString (String)	Returns the height and width of a given string as a Vector2 .
SpriteFont.MeasureString (StringBuilder)	Returns the height and width of a mutable string as a Vector2 .

See Also

Reference

[SpriteFont Class](#)

[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteFont.MeasureString Method (String)

Returns the height and width of a given string as a [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 MeasureString (  
    string text  
)
```

Parameters

text

The string to measure.

Return Value

The height and width, in pixels, of *text*, when it is rendered.

Exceptions

Exception type	Condition
ArgumentNullException	<i>text</i> cannot be null .
ArgumentException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information see How To: Extend the Font Description Processor to Support Additional Characters .

See Also

Reference

[SpriteFont Class](#)

[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteFont.MeasureString Method (StringBuilder)

Returns the height and width of a mutable string as a [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 MeasureString (
    StringBuilder text
)
```

Parameters

text

The mutable (read/write) string to measure.

Return Value

The height and width, in pixels, of *text*, when it is rendered.

Exceptions

Exception type	Condition
ArgumentNullException	<i>text</i> cannot be null .
ArgumentOutOfRangeException	There is a character in <i>text</i> that was not imported by the SpriteFont . You must include all the characters in the character regions in the sprite font file. For more information, see How To: Extend the Font Description Processor to Support Additional Characters .

Remarks

By accepting a [StringBuilder](#) argument in place of a [String](#) argument, this form supports measurement of dynamically formatted text.

See Also

Reference

[SpriteFont Class](#)





[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteFont Properties

Public Properties

	Name	Description
	Characters	Gets a collection of all the characters that are included in the font.
	DefaultCharacter	Gets or sets the default character for the font.
	LineSpacing	The line spacing, in pixels, of this font.
	Spacing	Gets or sets the spacing of the font characters.

See Also

Reference

[SpriteFont Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

SpriteFont.Characters Property

Gets a collection of all the characters that are included in the font.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<char> Characters { get; }
```

Property Value

The collection of all characters included in this font.

See Also

Reference

[SpriteFont Class](#)

[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteFont.DefaultCharacter Property

Gets or sets the default character for the font.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<char> DefaultCharacter { get; set; }
```

Property Value

The default character for this font.

Remarks

If set to anything other than null, this character will be automatically substituted any time an attempt is made to draw or measure characters that are not in the font.

If set to null, missing characters will produce an exception.

See Also

Reference

[SpriteFont Class](#)

[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteFont.LineSpacing Property

The line spacing, in pixels, of this font.

The line spacing is the vertical distance between the base lines of two consecutive lines of text. Thus, it includes the blank space between lines as well as the height of the characters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int LineSpacing { get; set; }
```

Property Value

The height of one line, in pixels.

See Also

Reference

[SpriteFont Class](#)

[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteFont.Spacing Property

Gets or sets the spacing of the font characters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Spacing { get; set; }
```

Property Value

The spacing, in pixels, of the font characters.

See Also

Reference

[SpriteFont Class](#)

[SpriteFont Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SpriteSortMode Enumeration

Defines sprite sort-rendering options.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum SpriteSortMode
```

Members

Member name	Description
BackToFront	Same as Deferred mode, except sprites are sorted by depth in back-to-front order prior to drawing. This procedure is recommended when drawing transparent sprites of varying depths.
Deferred	Sprites are not drawn until End is called. End will apply graphics device settings and draw all the sprites in one batch, in the same order calls to Draw were received. This mode allows Draw calls to two or more instances of SpriteBatch without introducing conflicting graphics device settings. SpriteBatch defaults to Deferred mode.
FrontToBack	Same as Deferred mode, except sprites are sorted by depth in front-to-back order prior to drawing. This procedure is recommended when drawing opaque sprites of varying depths.
Immediate	Begin will apply new graphics device settings, and sprites will be drawn within each Draw call. In Immediate mode there can only be one active SpriteBatch instance without introducing conflicting device settings. Immediate mode is faster than Deferred mode.
Texture	Same as Deferred mode, except sprites are sorted by texture prior to drawing. This can improve performance when drawing non-overlapping sprites of uniform depth.

See Also

Reference

[SpriteBatch.Begin Method](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StateBlock Class

Encapsulates render states.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class StateBlock : IDisposable
```

Remarks **StateBlock** objects must be recreated when the device is reset.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[StateBlock](#) Members

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






StateBlock Members

The following tables list the members exposed by the StateBlock type.









Public Constructors

Name	Description
 StateBlock	Initializes a new instance of the StateBlock class.





Public Properties

Name	Description
 GraphicsDevice	Gets the GraphicsDevice associated with this StateBlock .
 IsContentLost	Gets the current state of the content on a device.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Name	Gets or sets the name of the StateBlock .
 Tag	Gets or sets the resource manager tag for the StateBlock .



Public Methods

Name	Description
 Apply	Applies the state block to the current GraphicsDevice .
 Capture	Captures the current value of states that are included in a state block.
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs when content is about to be lost due to a device reset.
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 ContentLost	Occurs when resources are lost (for example, when the current device is lost).
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[StateBlock Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

StateBlock Constructor

Initializes a new instance of the [StateBlock](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StateBlock (  
    GraphicsDevice graphicsDevice  
)
```

Parameters

graphicsDevice

The device to associate with the state block.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
InvalidOperationException	Unable to create this StateBlock resource on the graphics device.

Remarks

StateBlock objects must be recreated when the device is reset.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[StateBlock](#) Class









[StateBlock](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace





Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock Methods

Public Methods

Name	Description
 Apply	Applies the state block to the current GraphicsDevice .
 Capture	Captures the current value of states that are included in a state block.
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_ContentLost	Occurs when content is about to be lost due to a device reset.
 raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[StateBlock Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

StateBlock.Apply Method

Applies the state block to the current [GraphicsDevice](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Apply ()
```

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Capture Method

Captures the current value of states that are included in a state block.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Capture ()
```

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
StateBlock.Dispose ()	Immediately releases the unmanaged resources used by this object.
StateBlock.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

StateBlock.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [StateBlock](#). The [Dispose](#) method leaves the [StateBlock](#) in an unusable state. After calling [Dispose](#), you must release all references to the [StateBlock](#) so the garbage collector can reclaim the memory that the [StateBlock](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [StateBlock](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [StateBlock](#) object's [Finalize](#) method.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [StateBlock](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.raise_ContentLost Method

Note

This method is available only when developing for Windows.

Occurs when content is about to be lost due to a device reset.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void raise_ContentLost (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

The source of this event.

value1

The event arguments that are associated with the action that raised the event.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[ContentLost](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

StateBlock.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the [Disposing](#) event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

StateBlock.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[StateBlock Class](#)






[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the GraphicsDevice associated with this StateBlock .
	IsContentLost	Gets the current state of the content on a device.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Name	Gets or sets the name of the StateBlock .
	Tag	Gets or sets the resource manager tag for the StateBlock .

See Also

Reference

[StateBlock Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

StateBlock.GraphicsDevice Property

Gets the [GraphicsDevice](#) associated with this [StateBlock](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The [GraphicsDevice](#) associated with this [StateBlock](#).

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.IsContentLost Property

Gets the current state of the content on a device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsContentLost { get; }
```

Property Value

true if content was lost due to device lost or similar event; otherwise **false**.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[ContentLost](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Name Property

Gets or sets the name of the [StateBlock](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of the [StateBlock](#).

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Tag Property

Gets or sets the resource manager tag for the [StateBlock](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource manager tag for the [StateBlock](#).

See Also

Reference

[StateBlock Class](#)



[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock Events

Public Events

	Name	Description
	Content Lost	Occurs when resources are lost (for example, when the current device is lost).
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

See Also

Reference

[StateBlock Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

StateBlock.ContentLost Event

Occurs when resources are lost (for example, when the current device is lost).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public virtual event EventHandler ContentLost
```

Remarks

When **ContentLost** fires, [IsContentLost](#) is set to **true**.

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[raise_ContentLost](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StateBlock.Disposing Event

Occurs when [Dispose](#) is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime (CLR).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[StateBlock Class](#)

[StateBlock Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

StencilOperation Enumeration

Defines stencil buffer operations.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum StencilOperation
```

Members

Member name	Description
Decrement	Decrements the stencil-buffer entry, wrapping to the maximum value if the new value is less than 0.
DecrementSaturation	Decrements the stencil-buffer entry, clamping to 0.
Increment	Increments the stencil-buffer entry, wrapping to 0 if the new value exceeds the maximum value.
IncrementSaturation	Increments the stencil-buffer entry, clamping to the maximum value.
Invert	Inverts the bits in the stencil-buffer entry.
Keep	Does not update the stencil-buffer entry. This is the default value.
Replace	Replaces the stencil-buffer entry with a reference value.
Zero	Sets the stencil-buffer entry to 0.

Remarks Stencil-buffer entries are integer values that range from 0 to $2n - 1$, where n is the bit depth of the stencil buffer.

See Also

Concepts

[What Is a Stencil Buffer?](#)

Reference

[RenderState.StencilPass Property](#)

[RenderState.StencilFail Property](#)

[RenderState.StencilDepthBufferFail Property](#)

[RenderState.CounterClockwiseStencilPass Property](#)

[RenderState.CounterClockwiseStencilFail Property](#)

[RenderState.CounterClockwiseStencilDepthBufferFail Property](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

SurfaceFormat Enumeration

Defines various types of surface formats.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum SurfaceFormat
```

Members

Member name	Description
Alpha8	(Unsigned format) 8-bit alpha only.
Bgr233	(Unsigned format) 8-bit BGR texture format using 2 bits for blue, 3 bits for green, and 3 bits for red.
Bgr24	(Unsigned format) 24-bit BGR pixel format with 8 bits per channel.
Bgr32	(Unsigned format) 32-bit BGR pixel format, where 8 bits are reserved for each color.
Bgr444	(Unsigned format) 16-bit BGR pixel format where 4 bits are reserved for each color.
Bgr555	(Unsigned format) 16-bit BGR pixel format where 5 bits are reserved for each color.
Bgr565	(Unsigned format) 16-bit BGR pixel format with 5 bits for blue, 6 bits for green, and 5 bits for red.
Bgra1010102	(Unsigned format) 32-bit pixel format using 10 bits each for blue, green, and red; and 2 bits for alpha.
Bgra2338	(Unsigned format) 16-bit BGRA format using 2 bits for blue, 3 bits each for red and green; and 8 bits for alpha.
Bgra4444	(Unsigned format) 16-bit BGRA pixel format with 4 bits for each channel.
Bgra5551	(Unsigned format) 16-bit BGRA pixel format where 5 bits are reserved for each color and 1 bit is reserved for alpha.
Color	(Unsigned format) 32-bit ARGB pixel format with alpha, using 8 bits per channel.

Depth15Stencil1	(Buffer format) 16-bit depth-buffer bit depth where 15 bits are reserved for the depth channel and 1 bit is reserved for the stencil channel.
Depth16	(Buffer format) 16-bit depth-buffer bit depth.
Depth24	(Buffer format) 32-bit depth-buffer bit depth using 24 bits for the depth channel.
Depth24Stencil4	(Buffer format) 32-bit depth-buffer bit depth using 24 bits for the depth channel and 4 bits for the stencil channel.
Depth24Stencil8	(Buffer format) 32-bit depth-buffer bit depth using 24 bits for the depth channel and 8 bits for the stencil channel.
Depth24Stencil8Single	(Buffer format) A non-lockable format that contains 24 bits of depth (in a 24-bit floating-point format – 20e4) and 8 bits of stencil.
Depth32	(Buffer format) 32-bit depth-buffer bit depth.
Dxt1	DXT1 compression texture format. The runtime will not allow an application to create a surface using a DXTn format unless the surface dimensions are multiples of 4. This applies to offscreen-plain surfaces, render targets, 2D textures, cube textures, and volume textures.
Dxt2	DXT2 compression texture format. The runtime will not allow an application to create a surface using a DXTn format unless the surface dimensions are multiples of 4. This applies to offscreen-plain surfaces, render targets, 2D textures, cube textures, and volume textures.
Dxt3	DXT3 compression texture format. The runtime will not allow an application to create a surface using a DXTn format unless the surface dimensions are multiples of 4. This applies to offscreen-plain surfaces, render targets, 2D textures, cube textures, and volume textures.
Dxt4	DXT4 compression texture format. The runtime will not allow an application to create a surface using a DXTn format unless the surface dimensions are multiples of 4. This applies to offscreen-plain surfaces, render targets, 2D textures, cube textures, and volume textures.
Dxt5	DXT5 compression texture format. The runtime will not allow an application to create a surface using a DXTn format unless the surface dimensions are multiples of 4. This applies to offscreen-plain surfaces, render targets, 2D textures, cube textures, and volume textures.

HalfSingle	(Floating-point format) 16-bit float format using 16 bits for the red channel.
HalfVector2	(Floating-point format) 32-bit float format using 16 bits for the red channel and 16 bits for the green channel.
HalfVector4	(Floating-point format) 64-bit float format using 16 bits for each channel (alpha, blue, green, red).
Luminance16	(Unsigned format) 16-bit luminance only.
Luminance8	(Unsigned format) 8-bit luminance only.
LuminanceAlpha16	(Unsigned format) 16-bit using 8 bits each for alpha and luminance.
LuminanceAlpha8	(Unsigned format) 8-bit using 4 bits each for alpha and luminance.
Multi2Bgra32	MultiElement texture (not compressed)
NormalizedAlpha1010102	(Mixed format) 32-bit bump-map format using 2 bits for alpha and 10 bits each for w, v, and u.
NormalizedByte2	(Signed format) 16-bit bump-map format using 8 bits each for u and v data.

Normalized DBYTE2C omputed	(Signed format) 16-bit normal compression format. The texture sampler computes the C channel from: $C = \sqrt{1 - U^2 - V^2}$.
Normalized DBYTE4	(Signed format) 32-bit bump-map format using 8 bits for each channel.
Normalized dLuminance16	(Mixed format) 16-bit bump-map format with luminance using 6 bits for luminance, and 5 bits each for v and u.
Normalized dLuminance32	(Mixed format) 32-bit bump-map format with luminance using 8 bits for each channel.
Normalized dShort2	(Signed format) 32-bit bump-map format using 16 bits for each channel.
Normalized dShort4	(Signed format) 64-bit bump-map format using 16 bits for each component.
Palette8	(Unsigned format) 8-bit color indexed.
PaletteAlpha16	(Unsigned format) 8-bit color indexed with 8 bits of alpha.

Rg32	(Unsigned format) 32-bit pixel format using 16 bits each for red and green.
Rgb32	(Unsigned format) 32-bit RGB pixel format, where 8 bits are reserved for each color.
Rgba1010102	(Unsigned format) 32-bit RGBA pixel format using 10 bits for each color and 2 bits for alpha.
Rgba32	(Unsigned format) 32-bit RGBA pixel format with alpha, using 8 bits per channel.
Rgba64	(Unsigned format) 64-bit RGBA pixel format using 16 bits for each component.
Single	(IEEE format) 32-bit float format using 32 bits for the red channel.
Unknown	Surface format is unknown.
Vector2	(IEEE format) 64-bit float format using 32 bits for the red channel and 32 bits for the green channel.
Vector4	(IEEE format) 128-bit float format using 32 bits for each channel (alpha, blue, green, red).
VideoRGB	A 16-bit packed RGB format analogous to VideoYuYv (U0Y0, V0Y1, U2Y2, and so on). It requires a pixel pair to properly represent the color value. The first pixel in the pair contains 8 bits of green (in the low 8 bits) and 8 bits of red (in the high 8 bits). The second pixel contains 8 bits of green (in the low 8 bits) and 8 bits of blue (in the high 8 bits). Together, the two pixels share the red and blue components, while each has a unique green component (R0G0, B0G1, R2G2, and so on). The texture sampler does not normalize the colors when looking up into a pixel shader; they remain in the range of 0.0f to 255.0f. This is true for all programmable pixel shader models. For the fixed-function pixel shader, the hardware should normalize to the 0.f to 1.f range and essentially treat it as the VideoYyVy texture. Hardware that exposes this format must have PixelShader1xMaxValue set to a value capable of handling that range.
VideoRGB	A 16-bit packed RGB format analogous to VideoUyVy (Y0U0, Y1V0, Y2U2, and so on). It requires a pixel pair to properly represent the color value. The first pixel in the pair contains 8 bits of green (in the high 8 bits) and 8 bits of red (in the low 8 bits). The second pixel contains 8 bits of green (in the high 8 bits) and 8 bits of blue (in the low 8 bits). Together, the two pixels share the red and blue components, while each has a unique green component (G0R0, G1B0, G2R2, and so on). The texture sampler does not normalize the colors when looking up into a pixel shader; they remain in the range of 0.0f to 255.0f. This is true for all programmable pixel shader models. For the fixed-function pixel shader, the hardware should normalize to the 0.f to 1.f range and essentially treat it as the VideoYyVy texture. Hardware that exposes this format must have PixelShader1xMaxValue set to a value capable of handling that range.
VideoUYVY	YUY2 format (PC98 compliance)
VideoUYVY	VideoYuYv format (PC98 compliance)

Remarks

In the XNA Framework, all two-dimensional (2D) images are represented by a range of memory called a surface. Within a surface, each element holds a color value representing a small section of the image, called a pixel. An image's detail level is defined by both the number of pixels needed to represent the image, and the number of bits needed for the image's color

spectrum. For example, an image that is 800 pixels wide by 600 pixels high with 32 bits of color for each pixel (written as 800×600×32) is more detailed than an image that is 640 pixels wide by 480 pixels tall with 16 bits of color for each pixel (written as 640×480×16). Likewise, the more detailed image requires a larger surface to store the data. For an 800×600×32 image, the surface's array dimensions are 800×600, and each element holds a 32-bit value to represent its color.

All formats are listed from left to right, most-significant bit to least-significant bit. For example, ARGB formats are ordered from the most-significant bit channel A (alpha), to the least-significant bit channel B (blue). When traversing surface data, the data is stored in memory from least-significant bit to most-significant bit, which means that the channel order in memory is from least-significant bit (blue) to most-significant bit (alpha).

The default value for formats that contain undefined channels (Rg32, Alpha8, and so on) is 1. The only exception is the Alpha8 format, which is initialized to 000 for the three color channels.

The order of the bits is from the most-significant byte first, so LuminanceAlpha16 indicates that the high byte of this 2-byte format is alpha. Depth16 indicates a 16-bit integer value and an application-lockable surface.

Floating-point formats are also known as s10e5 formats. IEEE formats are also known as s23e8 formats. Data in mixed formats can contain a combination of unsigned data and signed data. Data in a signed format can be both positive and negative. Signed formats use combinations of (U), (V), (W), and (Q) data. Data in an unsigned format must be positive. Unsigned formats use combinations of (R)ed, (G)reen, (B)lue, (A)lpha, (L)uminance, and (P)alette data. Palette data is also referred to as "color-indexed" data because the data is used to index a color palette.

Back-Buffer or Display Formats

The following formats are the only valid formats for a back buffer or a display.

Format	Back Buffer	Display
Bgra1010102	x	x (full-screen mode only)
Color	x	
Bgr32	x	x
Bgra5551	x	
Bgr555	x	x
Bgr565	x	x

Depth-stencil Buffer Formats

Depth-stencil buffers each have unique formats. All depth-stencil formats except **Depth16Lockable** indicate no particular bit ordering per pixel, and the driver is allowed to consume more than the indicated number of bits-per-depth channel (but not stencil channel).

⚠Caution
If the back buffer size has changed and the depth stencil must have the corresponding size, recreate the depth stencil when the device is reset.

DXT Compressed Texture Formats

The runtime will not allow an application to create a surface using a DXTn format unless the surface dimensions are multiples of 4. This applies to offscreen-plain surfaces, render targets, 2D textures, cube textures, and volume textures.

Direct3D Equivalents

A mapping of each Direct3D 9 surface format name to the **SurfaceFormat** equivalent is listed in the following table.

	Direct3D Surface Format	SurfaceFormat equivalent
Floating Point		
Float32	D3DFMT_R32F	Single
	D3DFMT_G32R32F	Vector2
	D3DFMT_A32B32G32R32F	Vector4
Float16	D3DFMT_R16F	HalfSingle
	D3DFMT_G16R16F	HalfVector2
	D3DFMT_A16B16G16R16F	HalfVector4
Unsigned Normalized		
64 bpp	D3DFMT_A16B16G16R16	Rgba64
32 bpp	D3DFMT_A8R8G8B8	Color
	D3DFMT_X8R8G8B8	Bgr32
	D3DFMT_A8B8G8R8	Rgba32

	D3DFMT_X8B8G8R8	Rgb32
	D3DFMT_A2R10G10B10	Bgra1010102
	D3DFMT_A2B10G10R10	Rgba1010102
	D3DFMT_G16R16	Rg32
24 bpp	D3DFMT_R8G8B8	Bgr24
16 bpp	D3DFMT_R5G6B5	Bgr565
	D3DFMT_A1R5G5B5	Bgra5551
	D3DFMT_X1R5G5B5	Bgr555
	D3DFMT_A4R4G4B4	Bgra4444
	D3DFMT_X4R4G4B4	Bgr444
	D3DFMT_A8R3G3B2	Bgra2338
8 bpp	D3DFMT_A8	Alpha8
	D3DFMT_R3G3B2	Bgr233
Signed Normalized		
	D3DFMT_V8U8	NormalizedByte2
	D3DFMT_Q8W8V8U8	NormalizedByte4
	D3DFMT_V16U16	NormalizedShort2
	D3DFMT_Q16W16V16U16	NormalizedShort4
	D3DFMT_CxV8U8	NormalizedByte2Computed
Block Compressed		
	D3DFMT_Dxt1	Dxt1
	D3DFMT_Dxt2	Dxt2
	D3DFMT_Dxt3	Dxt3
	D3DFMT_Dxt4	Dxt4
	D3DFMT_Dxt5	Dxt5
Luminance		
	D3DFMT_L8	Luminance8
	D3DFMT_L16	Luminance16
	D3DFMT_A4L4	LuminanceAlpha8
	D3DFMT_A8L8	LuminanceAlpha16
Palettized		
	D3DFMT_P8	Palette8
	D3DFMT_A8P8	PaletteAlpha16
Mixed Formats		
	D3DFMT_A2W10V10U10	NormalizedAlpha1010102
	D3DFMT_X8L8V8U8	NormalizedLuminance32
	D3DFMT_L6V5U5	NormalizedLuminance16
	D3DFMT_Yuy2	VideoUyVy
	D3DFMT_Uyvy	VideoYuYv
	D3DFMT_Multi2Argb8	Multi2Bgra32
	D3DFMT_R8G8B8G8	VideoGrGb
	D3DFMT_G8R8G8B8	VideoRgBg
Depth Formats		
	D3DFMT_D15S1	Depth15Stencil1
	D3DFMT_D16	Depth16
	D3DFMT_D24S8	Depth24Stencil8
	D3DFMT_D24SingleS8	Depth24Stencil8Single
	D3DFMT_D24X4S4	Depth24Stencil4
	D3DFMT_D24X8	Depth24
	D3DFMT_D32	Depth32

See Also

Concepts

[Xbox 360 Surface Formats](#)

Tasks

[How To: Create a Depth Texture](#)

Reference

[DisplayMode.Format](#) Property
[DisplayModeCollection.Item](#) Property
[GraphicsAdapter.CheckDepthStencilMatch](#) Method
[GraphicsAdapter.CheckDeviceFormat](#) Method
[GraphicsAdapter.CheckDeviceFormatConversion](#) Method
[GraphicsAdapter.CheckDeviceMultiSampleType](#) Method
[GraphicsAdapter.CheckDeviceType](#) Method
[PresentationParameters.BackBufferFormat](#) Property
[Texture2D](#) Constructor
[Texture2D.Format](#) Property
[Texture3D](#) Constructor
[Texture3D.Format](#) Property
[TextureCreationParameters](#) Constructor
[TextureCreationParameters.Format](#) Property
[TextureCube](#) Constructor
[TextureCube.Format](#) Property
[TextureInformation](#) Constructor
[TextureInformation.Format](#) Property
[Microsoft.Xna.Framework.Graphics](#) Namespace
Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SwapEffect Enumeration

Defines how the device front buffer and back buffer are to be swapped when [Present](#) is called.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum SwapEffect
```

Members

Member name	Description
Copy	<p>This swap effect can be specified only for a swap chain that comprises a single back buffer.</p> <p>Whether the swap chain is windowed or full-screen, the runtime guarantees the semantics implied by a copy-based Present operation; specifically, that the operation leaves the content of the back buffer unchanged, instead of replacing it with the content of the front buffer as a flip-based Present operation would.</p> <p>For a windowed swap chain, a Present operation causes the back buffer content to be copied immediately to the client area of the target window. No attempt is made to synchronize the copy with the vertical retrace period of the display adapter, so tearing effects may be observed.</p> <p>For a full-screen swap chain, the runtime uses a combination of flip and copy operations, which are supported by hidden back buffers if necessary, to accomplish the Present operation. Accordingly, the presentation is synchronized with the display adapter's vertical retrace and its rate is constrained by the chosen presentation interval. A swap chain specified with the PresentationInterval.Immediate flag is the only exception. (For more information, see PresentationInterval.) In this case, a Present operation copies the back buffer content directly to the front buffer without waiting for the vertical retrace.</p>
Default	<p>The default swap effect is Discard.</p>
Discard	<p>When a swap chain is created with a swap effect of Flip or Copy, the runtime guarantees that a Present operation will not affect the content of any of the back buffers. However, meeting this guarantee can involve substantial video memory or processing overheads, especially when implementing flip semantics for a windowed swap chain or copy semantics for a full-screen swap chain.</p> <p>An application can use the Discard swap effect to avoid these overheads and to enable the display driver to choose the most efficient presentation technique for the swap chain.</p> <p>Discard is also the only swap effect that can be used when specifying a value other than None for MultiSampleType. Like a swap chain that uses Flip, a swap chain that uses Discard might include more than one back buffer.</p> <p>The swap chain is essentially a queue where 0 always indexes the back buffer that will be displayed by the next Present operation and from which buffers are discarded once they have been displayed. An application that uses this swap effect should update an entire back buffer before invoking a Present operation that displays it.</p> <p>The debug version of the runtime overwrites the contents of discarded back buffers with random data, to enable developers to verify that their applications are updating the entire back buffer surface correctly.</p> <p>For a full-screen swap chain, the presentation rate is determined by the value assigned to PresentationInterval when the device or swap chain is created. Unless this value is PresentationInterval.Immediate, the presentation is synchronized with the vertical sync of the monitor. For a windowed swap chain, the presentation is implemented by means of copy operations, and always occurs immediately.</p>

Flip	<p>The swap chain might include multiple back buffers and is essentially a circular queue that includes the front buffer. Within this queue, the back buffers are always numbered sequentially from 0 to $(n - 1)$, where n is the number of back buffers, so that 0 denotes the least recently presented buffer.</p> <p>When Present is invoked, the queue is rotated so that the front buffer becomes the back buffer $(n - 1)$, while the back buffer 0 becomes the new front buffer.</p> <p>For a full-screen swap chain, the presentation rate is determined by the value assigned to the PresentationInterval when the device or swap chain is created. Unless this value is PresentInterval.Immediate, the presentation is synchronized with the vertical sync of the monitor.</p> <p>For a windowed swap chain, the flipping is implemented by means of copy operations, and the presentation always occurs immediately.</p>
-------------	---

Remarks

The state of the back buffer after a call to [Present](#) is well-defined by each of these swap effects, and whether the Microsoft Direct3D device was created with a full-screen swap chain or a windowed swap chain has no effect on this state. In particular, the **Flip** swap effect operates the same whether windowed or full-screen, and the Direct3D runtime guarantees this by creating extra buffers. As a result, it is recommended that applications use **Discard** whenever possible to avoid any performance penalties, because the current swap effect is always the most efficient in terms of memory consumption and performance.

Full-screen destination alpha will not work with applications that use **Flip** or **Discard**. This means that the **DestinationBlend** render state will not work as expected because, from the driver's point of view, full-screen swap chains with these swap effects do not have an explicit pixel format. The driver infers that it should take on the display format, which does not have an alpha channel. To work around this, use **Copy** and check [SupportsAlphaFullScreenFlipOrDiscard](#), which indicates whether the driver can perform alpha blending when **Flip** or **Discard** is used.

See Also

Reference

[PresentationParameters.SwapEffect](#) Property

[GraphicsDevice.Present](#) Method

[PresentationParameters.BackBufferCount](#) Property

[PresentationParameters.MultiSampleType](#) Property

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture Class

Represents a texture resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract class Texture : GraphicsResource
```

See Also

Tasks

[How To: Create Custom Texture Effects](#)

Reference

[Texture Members](#)









[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune












Texture Members

The following tables list the members exposed by the Texture type.



Public Properties

	Name	Description
	GraphicsDevice	(Inherited from GraphicsResource .)
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	Gets the number of texture levels in a multilevel texture.
	LevelOfDetail	Gets or sets the highest level-of-detail mipmap stored for a managed texture.
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Tag	(Inherited from GraphicsResource .)


Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
 S	FromFile	Overloaded. Creates a texture resource from a file.
	GenerateMipMaps	Request generation of mipmap sublevels for a render target texture.
 S	GetCreationParameters	Overloaded. Loads texture creation parameters from a file.
	GetHashCode	(Inherited from Object .)
 S	GetTextureInformation	Overloaded. Retrieves information about a given image file.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	Saves a texture to a file.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

Public Events

	Name	Description
	Disposing	(Inherited from GraphicsResource .)

See Also















Reference

[Texture Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
 	FromFile	Overloaded. Creates a texture resource from a file.
	GenerateMipMaps	Request generation of mipmap sublevels for a render target texture.
 	GetCreationParameters	Overloaded. Loads texture creation parameters from a file.
	GetHashCode	(Inherited from Object .)
 	GetTextureInformation	Overloaded. Retrieves information about a given image file.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	Saves a texture to a file.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[Texture Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
Texture.Dispose (Boolean)	Releases the unmanaged resources used by this object and optionally releases the managed resources.
Texture.Dispose ()	(Inherited from GraphicsResource .)
Texture.Dispose (Boolean)	(Inherited from GraphicsResource .)

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture.Dispose Method (Boolean)

Releases the unmanaged resources used by this object and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute\(U1\)](#)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [Texture](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture.FromFile Method

Creates a texture resource from a file.

Overload List

Name	Description
Texture.FromFile (GraphicsDevice, Stream)	Creates a texture resource from a stream.
Texture.FromFile (GraphicsDevice, Stream, Int32)	Creates a texture resource from a stream, specifying the number of bytes in the stream.
Texture.FromFile (GraphicsDevice, Stream, Int32, TextureCreationParameters)	Loads a Texture from a file.
Texture.FromFile (GraphicsDevice, Stream, TextureCreationParameters)	Loads a Texture from a file.
Texture.FromFile (GraphicsDevice, String)	Creates a texture resource from a file.
Texture.FromFile (GraphicsDevice, String, Int32, Int32, Int32)	Creates a texture resource from a file, specifying the width, height, and depth of the texture in pixels.
Texture.FromFile (GraphicsDevice, String, TextureCreationParameters)	Creates a texture resource from a file, specifying the parameters to be used in creation of the texture.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture.FromFile Method (GraphicsDevice, Stream)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream  
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream that contains the texture data.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture.FromFile Method (GraphicsDevice, Stream, Int32)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    int numberBytes
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream that contains the texture data.

numberBytes

The number of bytes in *textureStream*.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture.FromFile Method (GraphicsDevice, Stream, Int32, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Loads a [Texture](#) from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    int numberBytes,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) to associate the texture with.

textureStream

The stream to load the texture from.

numberBytes

The number of bytes in the stream.

creationParameters

Parameters to use when creating the [Texture](#).

Return Value

The created [Texture](#).

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.FromFile Method (GraphicsDevice, Stream, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Loads a [Texture](#) from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) to associate the texture with.

textureStream

The stream to load the texture from.

creationParameters

Parameters to use when creating the [Texture](#).

Return Value

The created [Texture](#).

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture.FromFile Method (GraphicsDevice, String)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (
    GraphicsDevice graphicsDevice,
    string filename
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.FromFile Method (GraphicsDevice, String, Int32, Int32, Int32)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file, specifying the width, height, and depth of the texture in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    int width,
    int height,
    int depth
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

width

The width, in pixels, of the texture.

height

The height, in pixels, of the texture.

depth

The depth, in pixels, of the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)
Platforms Windows XP SP2, Windows Vista

Texture.FromFile Method (GraphicsDevice, String, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file, specifying the parameters to be used in creation of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.GenerateMipMaps Method

Request generation of mipmap sublevels for a render target texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GenerateMipMaps (
    TextureFilter filterType
)
```

Parameters

filterType

Determines how the texture will be filtered for each mipmap level.

Exceptions

Exception type	Condition
NotSupportedException	GenerateMipMaps is not supported on Xbox 360.

Remarks

This method is valid only if [TextureUsage.AutoGenerateMipMap](#) is set when calling the constructor for a render target texture. Setting this flag indicates that mip levels should be generated, but not when they should be generated. For this case, **GenerateMipMaps** hints to the driver that it should generate sublevels if it has not done so already.

Only the top texture level is accessible to the application, and as a result [LevelCount](#) will not be changed after a call to **GenerateMipMaps**. The texture sublevels are not accessible since the driver creates them only when needed.

Example

In this example, a new [Texture2D](#) is created for use as a render target, the data is initialized with a call to [ResolveBackBuffer](#), and the generation of mipmaps for the resolved render target is requested.

C#

```
ResolveTexture2D renderTargetTexture;
renderTargetTexture = new ResolveTexture2D(
    graphics.GraphicsDevice,
    graphics.GraphicsDevice.PresentationParameters.BackBufferWidth,
    graphics.GraphicsDevice.PresentationParameters.BackBufferHeight,
    1,
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

graphics.GraphicsDevice.ResolveBackBuffer(renderTargetTexture);
renderTargetTexture.GenerateMipMaps( TextureFilter.Linear );
```

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture.GetCreationParameters Method

Loads texture creation parameters from a file.

Overload List

Name	Description
Texture.GetCreationParameters (GraphicsDevice, Stream)	Loads texture creation parameters from a stream.
Texture.GetCreationParameters (GraphicsDevice, Stream, Int32)	Loads texture creation parameters from a stream, specifying the number of bytes in the stream.
Texture.GetCreationParameters (GraphicsDevice, String)	Loads texture creation parameters from a file.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture.GetCreationParameters Method (GraphicsDevice, Stream)

Note

This method is available only when developing for Windows.

Loads texture creation parameters from a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCreationParameters GetCreationParameters (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) to associate with the texture creation parameters.

textureStream

Stream containing the texture creation data.

Return Value

The texture creation parameters.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture.GetCreationParameters Method (GraphicsDevice, Stream, Int32)

Note

This method is available only when developing for Windows.

Loads texture creation parameters from a stream, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCreationParameters GetCreationParameters (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream,  
    int numberBytes  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) to associate with the texture creation parameters.

textureStream

Stream containing the texture data.

numberBytes

The number of bytes in *textureStream*

Return Value

The texture creation parameters.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.GetCreationParameters Method (GraphicsDevice, String)

Note

This method is available only when developing for Windows.

Loads texture creation parameters from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCreationParameters GetCreationParameters (
    GraphicsDevice graphicsDevice,
    string filename
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) to associate with the texture creation parameters.

filename

The name of the file to load the creation parameters from.

Return Value

The texture creation parameters.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.GetTextureInformation Method

Retrieves information about a given image file.

Overload List

Name	Description
Texture.GetTextureInformation (Stream)	Retrieves information about a given image file stream.
Texture.GetTextureInformation (Stream, Int32)	Retrieves information about a given image file stream, specifying the number of bytes in the stream.
Texture.GetTextureInformation (String)	Retrieves information about a given image file.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture.GetTextureInformation Method (Stream)

Note

This method is available only when developing for Windows.

Retrieves information about a given image file stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureInformation GetTextureInformation (  
    Stream textureStream  
)
```

Parameters

textureStream

Stream containing the image data.

Return Value

A description of the data in the source file.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.GetTextureInformation Method (Stream, Int32)

Note

This method is available only when developing for Windows.

Retrieves information about a given image file stream, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureInformation GetTextureInformation (  
    Stream textureStream,  
    int numberBytes  
)
```

Parameters

textureStream

Stream containing the image data.

numberBytes

The number of bytes in *textureStream*.

Return Value

A description of the data in the source file.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.GetTextureInformation Method (String)

Note

This method is available only when developing for Windows.

Retrieves information about a given image file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureInformation GetTextureInformation (
    string filename
)
```

Parameters

filename

File name of image.

Return Value

A description of the data in the source file.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture.Save Method

Note

This method is available only when developing for Windows.

Saves a texture to a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Save (  
    string filename,  
    ImageFileFormat format  
)
```

Parameters

filename

The file name of the destination image.

format

The file format to use when saving. This function supports saving to all [ImageFileFormat](#) formats except Portable Pixmap (.ppm) and Targa/Truevision Graphics Adapter (.tga).

Remarks

This function handles conversion to and from compressed texture formats.

If the volume is nondynamic and located in video memory, **Save** will fail because this function cannot lock nondynamic volumes located in video memory.

See Also

Reference

[Texture Class](#)









[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture Properties

Public Properties

	Name	Description
	GraphicsDevice	(Inherited from GraphicsResource .)
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	Gets the number of texture levels in a multilevel texture.
	LevelOfDetail	Gets or sets the highest level-of-detail mipmap stored for a managed texture.
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Tag	(Inherited from GraphicsResource .)

See Also

Reference

[Texture Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture.LevelCount Property

Gets the number of texture levels in a multilevel texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int LevelCount { get; }
```

Property Value

The number of texture levels in the multilevel texture.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture.LevelOfDetail Property

Gets or sets the highest level-of-detail mipmap stored for a managed texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int LevelOfDetail { get; set; }
```

Property Value

The highest level-of-detail mipmap stored for a managed texture. The default value is 0, causing loading of all mipmapped textures.

Remarks

For example, a 512×512 texture has the following dimensions for each mipmap level: level 0 is 512×512, level 1 is 256×256, level 2 is 128×128, level 3 is 64×64, and so on.

For this example, a value of 3 indicates that the driver only loads mipmaps of size 64×64 and smaller into video memory.

This member is ignored for games targeting the Xbox 360 platform because copying of texture data from system memory to video memory is not permitted.

See Also

Reference

[Texture Class](#)

[Texture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D Class

Represents a 2D grid of texels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class Texture2D : Texture
```

Remarks

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A [Texture2D](#) resource contains a 2D grid of texels. Each texel is addressable by a u, v vector. Since it is a texture resource, it may contain mipmap levels. Figure 1 shows a fully populated 2D texture resource.



Figure 1. Texture2D Resource Architecture

This texture resource contains a single 3×5 texture with three mipmap levels.

See Also

Tasks

[How To: Create Custom Texture Effects](#)

Reference

[Texture2D Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune













Texture2D Members

The following tables list the members exposed by the Texture2D type.















Public Constructors

Name	Description
 Texture2D	Overloaded. Creates an uninitialized Texture2D resource of the specified dimensions. To initialize a Texture2D from an existing file, see the static method ContentManager.Load or Texture2D.FromFile .




Public Properties

Name	Description
 Format	Gets the pixel format of this texture resource.
 GraphicsDevice	(Inherited from GraphicsResource .)
 Height	Gets the height of this texture resource, in pixels.
 IsDisposed	(Inherited from GraphicsResource .)
 LevelCount	(Inherited from Texture .)
 LevelOfDetail	(Inherited from Texture .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 Tag	(Inherited from GraphicsResource .)
 TextureUsage	Gets the state of the related TextureUsage enumeration.
 Width	Gets the width of this texture resource, in pixels.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
  FromFile	Overloaded. Creates a texture resource from a file.
 GenerateMipMaps	(Inherited from Texture .)
 GetCreationParameters	(Inherited from Texture .)
 GetData	Overloaded. Copies texture data into an array.
 GetHashCode	(Inherited from Object .)
 GetTextureInformation	(Inherited from Texture .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Save	(Inherited from Texture .)
 SetData	Overloaded. Copies array data to the texture.
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[Texture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D Constructor

Creates an uninitialized [Texture2D](#) resource of the specified dimensions. To initialize a [Texture2D](#) from an existing file, see the static method [ContentManager.Load](#) or [Texture2D.FromFile](#).

Overload List

Name	Description
Texture2D (GraphicsDevice, Int32, Int32)	Creates an uninitialized Texture2D resource of the specified dimensions. To initialize a Texture2D from an existing file, see the static method ContentManager.Load or Texture2D.FromFile .
Texture2D (GraphicsDevice, Int32, Int32, Int32, TextureUsage, SurfaceFormat)	Creates an uninitialized Texture2D resource of the specified dimensions. To initialize a Texture2D from an existing file, see the static method ContentManager.Load or Texture2D.FromFile .

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D Constructor (GraphicsDevice, Int32, Int32)

Creates an uninitialized [Texture2D](#) resource of the specified dimensions. To initialize a [Texture2D](#) from an existing file, see the static method [ContentManager.Load](#) or [Texture2D.FromFile](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D (
    GraphicsDevice graphicsDevice,
    int width,
    int height
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) used to display the texture.

width

The width of the texture, in pixels. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. If this value is 0, a value of 1 is used.

height

The height of the texture, in pixels. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. If this value is 0, a value of 1 is used.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>width</i> is larger than MaxTextureWidth. <i>height</i> is larger than MaxTextureHeight. <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels. <i>usage</i> is not a valid value. Texture resource can be created only with the following TextureUsage options: TextureUsage.AutoGenerateMipMap, TextureUsage.Tiled, and TextureUsage.Linear. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <p>The parameters specified are not compatible:</p> <ul style="list-style-type: none"> <i>format</i> and <i>numberLevels</i> are incompatible. Multi-element textures of type SurfaceFormat.Multi2Bgra32 cannot be mipmapped, so you need to set <i>numberLevels</i> to 1. <i>usage</i> and <i>numberLevels</i> are incompatible. Textures specifying the usage TextureUsage.AutoGenerateMipMap must set <i>numberLevels</i> to either 0 or 1. The <i>format</i> and <i>usage</i> requested are not supported by the graphics device. The device does not support creating a texture of the given format with the given TextureUsage.
ArgumentOutOfRangeException	<i>width</i> or <i>height</i> is less than or equal to zero; <i>width</i> and <i>height</i> must be greater than zero.
OutOfVideoMemoryException	Unable to create this resource on the graphics device.

Remarks

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A [Texture2D](#) resource contains a 2D grid of texels. Each texel is addressable by a u, v vector. Since it is a texture resource, it may contain mipmap levels. Figure 1 shows a fully populated 2D texture resource.

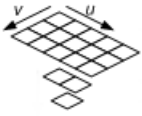


Figure 1. Texture2D Resource Architecture

This texture resource contains a single 3×5 texture with three mipmap levels.

Example

In this example, a new [Texture2D](#) is created for use as a render target, the data is initialized with a call to [ResolveBackBuffer](#), and the generation of mipmaps for the resolved render target is requested.

C#

```
ResolveTexture2D renderTargetTexture;  
renderTargetTexture = new ResolveTexture2D(  
    graphics.GraphicsDevice,  
    graphics.GraphicsDevice.PresentationParameters.BackBufferWidth,  
    graphics.GraphicsDevice.PresentationParameters.BackBufferHeight,  
    1,  
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);  
  
graphics.GraphicsDevice.ResolveBackBuffer(renderTargetTexture);  
renderTargetTexture.GenerateMipMaps( TextureFilter.Linear );
```

See Also

Reference

[ContentManager.Load Generic Method](#)

[Texture2D.FromFile Method](#)

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D Constructor (GraphicsDevice, Int32, Int32, Int32, TextureUsage, SurfaceFormat)

Creates an uninitialized [Texture2D](#) resource of the specified dimensions. To initialize a [Texture2D](#) from an existing file, see the static method [ContentManager.Load](#) or [Texture2D.FromFile](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    int numberLevels,
    TextureUsage usage,
    SurfaceFormat format
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) used to display the texture.

width

The width of the texture, in pixels. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. If this value is 0, a value of 1 is used.

height

The height of the texture, in pixels. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. If this value is 0, a value of 1 is used.

numberLevels

The number of levels in the texture. These smaller versions of the texture, known as mipmap levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If *numberLevels* is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures. Use [LevelCount](#) to see the number of levels generated.

usage

Options identifying the behaviors of this texture resource.

format

A [SurfaceFormat](#) value specifying the requested pixel format for the texture. The returned texture may be of a different format if the device does not support the requested format. Applications should check the format of the returned texture to ensure that it matches the requested format.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .

ArgumentException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • <i>width</i> is larger than MaxTextureWidth. • <i>height</i> is larger than MaxTextureHeight. • <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels. • <i>usage</i> is not a valid value. Texture resource can be created only with the following TextureUsage options: TextureUsage.AutoGenerateMipMap, TextureUsage.Tiled, and TextureUsage.Linear. • <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <p>The parameters specified are not compatible:</p> <ul style="list-style-type: none"> • <i>format</i> and <i>numberLevels</i> are incompatible. Multi-element textures of type SurfaceFormat.Multi2Bgra32 cannot be mipmapped, so you need to set <i>numberLevels</i> to 1. • <i>usage</i> and <i>numberLevels</i> are incompatible. Textures specifying the usage TextureUsage.AutoGenerateMipMap must set <i>numberLevels</i> to either 0 or 1. • The <i>format</i> and <i>usage</i> requested are not supported by the graphics device. The device does not support creating a texture of the given format with the given TextureUsage.
ArgumentOutOfRangeException	<i>width</i> or <i>height</i> is less than or equal to zero; <i>width</i> and <i>height</i> must be greater than zero.
OutOfVideoMemoryException	Unable to create this resource on the graphics device.

Remarks

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A [Texture2D](#) resource contains a 2D grid of texels. Each texel is addressable by a u, v vector. Since it is a texture resource, it may contain mipmap levels. Figure 1 shows a fully populated 2D texture resource.

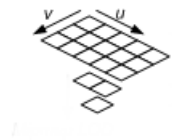


Figure 1. Texture2D Resource Architecture

This texture resource contains a single 3×5 texture with three mipmap levels.

Example

In this example, a new [Texture2D](#) is created for use as a render target, the data is initialized with a call to [ResolveBackBuffer](#), and the generation of mipmaps for the resolved render target is requested.

C#

```
ResolveTexture2D renderTargetTexture;
renderTargetTexture = new ResolveTexture2D(
    graphics.GraphicsDevice,
    graphics.GraphicsDevice.PresentationParameters.BackBufferWidth,
    graphics.GraphicsDevice.PresentationParameters.BackBufferHeight,
    1,
    graphics.GraphicsDevice.PresentationParameters.BackBufferFormat);

graphics.GraphicsDevice.ResolveBackBuffer(renderTargetTexture);
renderTargetTexture.GenerateMipMaps( TextureFilter.Linear );
```

See Also

Reference

[ContentManager.Load Generic Method](#)

[Texture2D.FromFile Method](#)

[Texture2D Class](#)















[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
 	FromFile	Overloaded. Creates a texture resource from a file.
	GenerateMipMaps	(Inherited from Texture .)
	GetCreationParameters	(Inherited from Texture .)
	GetData	Overloaded. Copies texture data into an array.
	GetHashCode	(Inherited from Object .)
	GetTextureInformation	(Inherited from Texture .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	(Inherited from Texture .)
	SetData	Overloaded. Copies array data to the texture.
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[Texture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
Texture2D.Dispose (Boolean)	Releases the unmanaged resources used by the Texture2D and optionally releases the managed resources.
Texture2D.Dispose ()	(Inherited from GraphicsResource .)

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D.Dispose Method (Boolean)

Releases the unmanaged resources used by the [Texture2D](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[[MarshalAsAttribute](#)(U1)] **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [Texture2D](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.FromFile Method

Creates a texture resource from a file.

Overload List

Name	Description
Texture2D.FromFile (GraphicsDevice, Stream)	Creates a texture resource from a stream.
Texture2D.FromFile (GraphicsDevice, Stream, Int32)	Creates a texture resource from a stream, specifying the number of bytes in the stream.
Texture2D.FromFile (GraphicsDevice, Stream, Int32, TextureCreationParameters)	Creates a texture resource from a stream, specifying the number of bytes in the stream and the texture creation parameters to use.
Texture2D.FromFile (GraphicsDevice, Stream, TextureCreationParameters)	Creates a texture resource from a stream, specifying the parameters to be used in creation of the texture.
Texture2D.FromFile (GraphicsDevice, String)	Creates a texture resource from a file.
Texture2D.FromFile (GraphicsDevice, String, Int32, Int32)	Creates a texture resource from a file, specifying the width and height of the texture in pixels.
Texture2D.FromFile (GraphicsDevice, String, TextureCreationParameters)	Creates a texture resource from a file, specifying the parameters to be used in creation of the texture.
Texture2D.FromFile (GraphicsDevice, String, Int32, Int32, Int32)	(Inherited from Texture .)

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D.FromFile Method (GraphicsDevice, Stream)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream  
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture2D.FromFile Method (GraphicsDevice, Stream, Int32)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    int numberBytes
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

numberBytes

The number of bytes in *textureStream*.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture2D.FromFile Method (GraphicsDevice, Stream, Int32, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the number of bytes in the stream and the texture creation parameters to use.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    int numberBytes,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the volume texture data.

numberBytes

The number of bytes in *textureStream*.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture2D.FromFile Method (GraphicsDevice, Stream, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the parameters to be used in creation of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture2D.FromFile Method (GraphicsDevice, String)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (
    GraphicsDevice graphicsDevice,
    string filename
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture2D.FromFile Method (GraphicsDevice, String, Int32, Int32)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file, specifying the width and height of the texture in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    int width,
    int height
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

width

The width, in pixels, of the texture.

height

The height, in pixels, of the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture2D.FromFile Method (GraphicsDevice, String, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file, specifying the parameters to be used in creation of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture2D FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture2D.GetData Method

Copies texture data into an array.

Overload List

Name	Description
Texture2D.GetData (Int32, Nullable<Rectangle>, T[], Int32, Int32)	Copies texture data into an array.
Texture2D.GetData (T[])	Copies texture data into an array.
Texture2D.GetData (T[], Int32, Int32)	Copies texture data into an array.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D.GetData Generic Method (Int32, Nullable<Rectangle>, T[], Int32, Int32)

Copies texture data into an array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    int level,
    Nullable<Rectangle> rect,
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array. This is usually a [Microsoft.Xna.Framework.Graphics.PackedVector](#) type which matches the [Format](#) of the texture, but it can be any fundamental type that matches the bit size of the surface format.

Parameters

level

The mipmap level to copy from.

rect

The section of the texture to copy. **null** indicates the data will be copied from the entire texture.

data

The array to receive texture data.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> must be of sufficient length to receive the data.
InvalidOperationException	The vertex stride is larger than the vertex buffer, or the vertex stride is too small for the type of data requested.

Example

In this example, the color of the pixel beneath the mouse is retrieved from the back buffer.

C#

```
if (Mouse.GetState().LeftButton == ButtonState.Pressed &&
    // If the left button is pressed
    Mouse.GetState().X > 0 && Mouse.GetState().Y > 0 &&
    // and we are inside the game window
    (Mouse.GetState().X <
     GraphicsDevice.PresentationParameters.BackBufferWidth) &&
    (Mouse.GetState().Y <
     GraphicsDevice.PresentationParameters.BackBufferHeight))
{
    backBufferData = new ResolveTexture2D(
        GraphicsDevice,
        GraphicsDevice.PresentationParameters.BackBufferWidth,
        GraphicsDevice.PresentationParameters.BackBufferHeight,
        1,
        GraphicsDevice.PresentationParameters.BackBufferFormat);
}
```

```
Rectangle sourceRectangle =  
    new Rectangle(Mouse.GetState().X, Mouse.GetState().Y, 1, 1);  
  
Color[] retrievedColor = new Color[1];  
  
GraphicsDevice.ResolveBackBuffer(backBufferData);  
  
backBufferData.GetData<Color>(0,  
    sourceRectangle,  
    retrievedColor,  
    0,  
    1);  
}
```

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.GetData Generic Method (T[])

Copies texture data into an array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    T[] data
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array. This is usually a [Microsoft.Xna.Framework.Graphics.PackedVector](#) type that matches the [Format](#) of the texture, but it can be any fundamental type that matches the bit size of the surface format.

Parameters

data

The array to receive texture data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> must be of sufficient length to receive the data.
InvalidOperationException	The vertex stride is larger than the vertex buffer, or the vertex stride is too small for the type of data requested.

Example

In this example, the color of the pixel beneath the mouse is retrieved from the back buffer.

C#

```
if (Mouse.GetState().LeftButton == ButtonState.Pressed &&
    // If the left button is pressed
    Mouse.GetState().X > 0 && Mouse.GetState().Y > 0 &&
    // and we are inside the game window
    (Mouse.GetState().X <
    GraphicsDevice.PresentationParameters.BackBufferWidth) &&
    (Mouse.GetState().Y <
    GraphicsDevice.PresentationParameters.BackBufferHeight))
{
    backBufferData = new ResolveTexture2D(
        GraphicsDevice,
        GraphicsDevice.PresentationParameters.BackBufferWidth,
        GraphicsDevice.PresentationParameters.BackBufferHeight,
        1,
        GraphicsDevice.PresentationParameters.BackBufferFormat);

    Rectangle sourceRectangle =
        new Rectangle(Mouse.GetState().X, Mouse.GetState().Y, 1, 1);

    Color[] retrievedColor = new Color[1];

    GraphicsDevice.ResolveBackBuffer(backBufferData);

    backBufferData.GetData<Color>(
        0,
        sourceRectangle,
        retrievedColor,
        0,
        1);
}
```

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.GetData Generic Method (T[], Int32, Int32)

Copies texture data into an array.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array. This is usually a [Microsoft.Xna.Framework.Graphics.PackedVector](#) type that matches the [Format](#) of the texture, but it can be any fundamental type that matches the bit size of the surface format.

Parameters

data

The array to receive texture data.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> must be of sufficient length to receive the data.
InvalidOperationException	The vertex stride is larger than the vertex buffer, or the vertex stride is too small for the type of data requested.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

Example

In this example, the color of the pixel beneath the mouse is retrieved from the back buffer.

C#

```
if (Mouse.GetState().LeftButton == ButtonState.Pressed &&
    // If the left button is pressed
    Mouse.GetState().X > 0 && Mouse.GetState().Y > 0 &&
    // and we are inside the game window
    (Mouse.GetState().X <
    GraphicsDevice.PresentationParameters.BackBufferWidth) &&
    (Mouse.GetState().Y <
    GraphicsDevice.PresentationParameters.BackBufferHeight))
{
    backBufferData = new ResolveTexture2D(
        GraphicsDevice,
        GraphicsDevice.PresentationParameters.BackBufferWidth,
        GraphicsDevice.PresentationParameters.BackBufferHeight,
        1,
        GraphicsDevice.PresentationParameters.BackBufferFormat);

    Rectangle sourceRectangle =
        new Rectangle(Mouse.GetState().X, Mouse.GetState().Y, 1, 1);

    Color[] retrievedColor = new Color[1];

    GraphicsDevice.ResolveBackBuffer(backBufferData);
```

```
backBufferData.GetData<Color>(
    0,
    sourceRectangle,
    retrievedColor,
    0,
    1);
}
```

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.SetData Method

Copies array data to the texture.

Overload List

Name	Description
Texture2D.SetData (Int32, Nullable<Rectangle>, T[], Int32, Int32, SetDataOptions)	Copies array data to the texture.
Texture2D.SetData (T[])	Copies array data to the texture at mipmap level 0.
Texture2D.SetData (T[], Int32, Int32, SetDataOptions)	Copies array data to the texture at mipmap level 0.

Exceptions

Exception type	Condition
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D.SetData Generic Method (Int32, Nullable<Rectangle>, T[], Int32, Int32, SetDataOptions)

Copies array data to the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    int level,
    Nullable<Rectangle> rect,
    T[] data,
    int startIndex,
    int elementCount,
    SetDataOptions options
) where T : ValueType
```

Type Parameters

T

The type of data to set.

Parameters

level

The mipmap level where the data will be placed.

rect

The section of the texture where the data will be placed. **null** indicates the data will be copied over the entire texture.

data

The array of data to copy. If *rect* is null, the number of elements in the array must be equal to the size of the texture, which is **Width** × **Height**; otherwise, the number of elements in the array should equal the size of the rectangle specified.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

options

Option that specifies whether existing data in the buffer will be kept after this operation.

Exceptions

Exception type	Condition
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

Example

This example demonstrates the use of the *rect* argument to set a single pixel in a texture.

C#

```
texture2D.SetData<Color>(
    0,
    new Rectangle(randomX, randomY, 1, 1),
    new Color[] { Color.White },
    0,
    1,
    SetDataOptions.None
);
```

See Also

Concepts

[Predicated Tiling](#)

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.SetData Generic Method (T[])

Copies array data to the texture at mipmap level 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    T[] data
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy. The number of elements in the array must be equal to the size of the texture, which is [Width](#) × [Height](#).

Exceptions

Exception type	Condition
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

Remarks

⚠Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Concepts

[Predicated Tiling](#)

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.SetData Generic Method (T[], Int32, Int32, SetDataOptions)

Copies array data to the texture at mipmap level 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    T[] data,
    int startIndex,
    int elementCount,
    SetDataOptions options
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array. Generally the type for this array will allow for elements in the same size as specified for the [Format](#) property of the texture.

Parameters

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy. This must be equal to the size of the texture, which is [Width](#) × [Height](#).

options

Option that specifies whether existing data in the buffer will be kept after this operation.

Exceptions

Exception type	Condition
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[Texture2D Class](#)













[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D Properties

Public Properties

	Name	Description
	Format	Gets the pixel format of this texture resource.
	GraphicsDevice	(Inherited from GraphicsResource .)
	Height	Gets the height of this texture resource, in pixels.
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	(Inherited from Texture .)
	LevelOfDetail	(Inherited from Texture .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Tag	(Inherited from GraphicsResource .)
	TextureUsage	Gets the state of the related TextureUsage enumeration.
	Width	Gets the width of this texture resource, in pixels.

See Also

Reference

[Texture2D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture2D.Format Property

Gets the pixel format of this texture resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; }
```

Property Value

The pixel format of this texture resource.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.Height Property

Gets the height of this texture resource, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

The height of this texture resource, in pixels.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.TextureUsage Property

Gets the state of the related [TextureUsage](#) enumeration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureUsage TextureUsage { get; }
```

Property Value

Indicates how the application uses the texture.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture2D.Width Property

Gets the width of this texture resource, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

The width of this texture resource, in pixels.

See Also

Reference

[Texture2D Class](#)

[Texture2D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Texture3D Class

Represents a 3D volume of texels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class Texture3D : Texture
```

Remarks

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A **Texture3D** resource (also known as a volume texture) contains a 3D volume of texels. Since it is a texture resource, it may contain mipmap levels. Figure 1 shows a fully populated **Texture3D** resource.

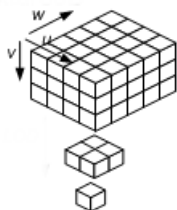


Figure 1. Texture3D Resource Architecture

When a **Texture3D** mipmap slice is bound as a render target output (by creating a [RenderTargetCube](#)), the **Texture3D** behaves identically to an array of [Texture2D](#) objects with n array slices, where n is the depth (third dimension) of the **Texture3D**.

See Also

Reference

[RenderTargetCube Class](#)

[Texture3D Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista














Texture3D Members

The following tables list the members exposed by the Texture3D type.















Public Constructors

	Name	Description
	Texture3D	Creates an uninitialized volume texture resource of the given dimensions, specifying the memory management mode for the resource. To initialize a Texture3D from an existing file, see the static method ContentManager.Load or FromFile .




Public Properties

	Name	Description
	Depth	Gets the depth of this volume texture resource, in pixels.
	Format	Gets the pixel format for this texture resource.
	GraphicsDevice	(Inherited from GraphicsResource .)
	Height	Gets the height of this texture resource, in pixels.
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	(Inherited from Texture .)
	LevelOfDetail	(Inherited from Texture .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Tag	(Inherited from GraphicsResource .)
	TextureUsage	Gets the state of the related TextureUsage enumeration.
	Width	Gets the width of this texture resource, in pixels.


Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	 FromFile	Overloaded. Creates a volume texture resource from a file.
	GenerateMipMaps	(Inherited from Texture .)
	GetCreationParameters	(Inherited from Texture .)
	GetData	Overloaded. Gets a copy of the texture data.
	GetHashCode	(Inherited from Object .)
	GetTextureInformation	(Inherited from Texture .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	(Inherited from Texture .)
	SetData	Overloaded. Copies array data to the texture.
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

Public Events

	Name	Description
	Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[Texture3D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D Constructor

Creates an uninitialized volume texture resource of the given dimensions, specifying the memory management mode for the resource. To initialize a [Texture3D](#) from an existing file, see the static method [ContentManager.Load](#) or [FromFile](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture3D (
    GraphicsDevice graphicsDevice,
    int width,
    int height,
    int depth,
    int numberLevels,
    TextureUsage usage,
    SurfaceFormat format
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) that will display the volume texture.

width

The width, in pixels, of the top-level of the volume texture. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. The pixel dimensions of subsequent levels will be the truncated value of half of the previous level's pixel dimension (independently). Each dimension clamps at a size of 1 pixel. Thus, if the division by two results in 0 (zero), 1 will be taken instead. The maximum dimension that a driver supports (for *width*, *height*, and *depth*) can be found in [MaxVolumeExtent](#).

height

The height, in pixels, of the top-level of the volume texture. This value must be a power of two if the [RequiresPower2](#) property of *graphicsDevice* is **true**. The pixel dimensions of subsequent levels will be the truncated value of half of the previous level's pixel dimension (independently). Each dimension clamps at a size of 1 pixel. Thus, if the division by two results in 0 (zero), 1 will be taken instead. The maximum dimension that a driver supports (for *width*, *height*, and *depth*) can be found in [MaxVolumeExtent](#).

depth

The depth, in pixels, of the top-level of the volume texture. This value must be a power of two if the [RequiresVolumeMapPower2](#) property of *graphicsDevice* is **true**. The pixel dimensions of subsequent levels will be the truncated value of half of the previous level's pixel dimension (independently). Each dimension clamps at a size of 1 pixel. Thus, if the division by two results in 0 (zero), 1 will be taken instead. The maximum dimension that a driver supports (for *width*, *height*, and *depth*) can be found in [MaxVolumeExtent](#).

numberLevels

The number of downsampled surfaces to create when preprocessing the texture. These smaller versions of the texture, known as mipmap levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If *numberLevels* is zero, all texture sublevels down to 1×1 pixels will be generated for hardware that supports mipmapped textures. Use [LevelCount](#) to see the number of levels generated.

usage

Options identifying the behaviors of this texture resource.

format

The format of all levels in the volume texture resource.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>width</i> , <i>height</i> , or <i>depth</i> is less than or equal to zero. <i>width</i> , <i>height</i> and <i>depth</i> must be greater than zero.

ArgumentException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • <i>numberLevels</i> is too large for a texture cube of this size. • <i>numberLevels</i> is greater than 32. Textures have a maximum of 32 levels. • <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. • <i>usage</i> is not a valid value. Texture resources can only be created with the following TextureUsage options: TextureUsage.AutoGenerateMipMap, TextureUsage.Tiled, and TextureUsage.Linear. • The combination of <i>width</i>, <i>height</i>, and <i>depth</i> requested results in a resource that is too large. • <i>format</i> and <i>numberLevels</i> are incompatible. Multi-element textures of type SurfaceFormat.Multi2Bgra32 cannot be mipmapped, set <i>numberLevels</i> to 1. • The graphics device does not support the requested <i>format</i> and <i>usage</i>. The device does not support creating a texture of the given format with the given TextureUsage.
OutOfVideoMemoryException	Unable to create this resource on the graphics device.

Remarks

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A **Texture3D** resource (also known as a volume texture) contains a 3D volume of texels. Since it is a texture resource, it may contain mipmap levels. Figure 1 shows a fully populated **Texture3D** resource.

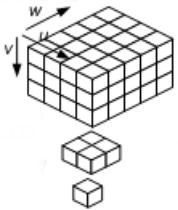


Figure 1. Texture3D Resource Architecture

When a **Texture3D** mipmap slice is bound as a render target output (by creating a [RenderTargetCube](#)), the **Texture3D** behaves identically to an array of [Texture2D](#) objects with *n* array slices, where *n* is the depth (third dimension) of the **Texture3D**.

See Also

Reference

[ContentManager.Load Generic Method](#)

[Texture3D.FromFile Method](#)

[Texture3D Class](#)















[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
 	FromFile	Overloaded. Creates a volume texture resource from a file.
	GenerateMipMaps	(Inherited from Texture .)
	GetCreationParameters	(Inherited from Texture .)
	GetData	Overloaded. Gets a copy of the texture data.
	GetHashCode	(Inherited from Object .)
	GetTextureInformation	(Inherited from Texture .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	(Inherited from Texture .)
	SetData	Overloaded. Copies array data to the texture.
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[Texture3D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
Texture3D.Dispose (Boolean)	Releases the unmanaged resources used by the Texture3D and optionally releases the managed resources.
Texture3D.Dispose ()	(Inherited from GraphicsResource .)

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D.Dispose Method (Boolean)

Releases the unmanaged resources used by the [Texture3D](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [Texture3D](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.FromFile Method

Creates a volume texture resource from a file.

Overload List

Name	Description
Texture3D.FromFile (GraphicsDevice, Stream)	Creates a volume texture resource from a stream.
Texture3D.FromFile (GraphicsDevice, Stream, Int32)	Creates a volume texture resource from a stream, specifying the number of bytes in the stream.
Texture3D.FromFile (GraphicsDevice, Stream, Int32, TextureCreationParameters)	Creates a volume texture resource from a stream, specifying the number of bytes in the stream and the texture creation parameters to use.
Texture3D.FromFile (GraphicsDevice, Stream, TextureCreationParameters)	Creates a volume texture resource from a stream, specifying the parameters to use for the texture creation.
Texture3D.FromFile (GraphicsDevice, String)	Creates a volume texture resource from a file.
Texture3D.FromFile (GraphicsDevice, String, Int32, Int32, Int32)	Creates a volume texture resource from a file, specifying the width, height, and depth in pixels.
Texture3D.FromFile (GraphicsDevice, String, TextureCreationParameters)	Creates a volume texture resource from a file, specifying the parameters to be used in creation of the texture.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D.FromFile Method (GraphicsDevice, Stream)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream  
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the volume texture data.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture3D.FromFile Method (GraphicsDevice, Stream, Int32)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a stream, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream,  
    int numberBytes  
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

numberBytes

The number of bytes in *textureStream*.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture3D.FromFile Method (GraphicsDevice, Stream, Int32, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a stream, specifying the number of bytes in the stream and the texture creation parameters to use.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    int numberBytes,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the volume texture data.

numberBytes

The number of bytes in *textureStream*.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture3D.FromFile Method (GraphicsDevice, Stream, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a stream, specifying the parameters to use for the texture creation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

creationParameters

Texture creation options.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture3D.FromFile Method (GraphicsDevice, String)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (
    GraphicsDevice graphicsDevice,
    string filename
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture3D.FromFile Method (GraphicsDevice, String, Int32, Int32, Int32)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a file, specifying the width, height, and depth in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    int width,
    int height,
    int depth
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

width

The width, in pixels, of the texture.

height

The height, in pixels, of the texture.

depth

The depth, in pixels, of the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)
Platforms Windows XP SP2, Windows Vista

Texture3D.FromFile Method (GraphicsDevice, String, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a volume texture resource from a file, specifying the parameters to be used in creation of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static Texture3D FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture3D.GetData Method

Gets a copy of the texture data.

Overload List

Name	Description
Texture3D.GetData (Int32, Int32, Int32, Int32, Int32, Int32, Int32, T[], Int32, Int32)	Gets a copy of the texture data, specifying the level and dimensions of the volume texture to copy.
Texture3D.GetData (T[])	Gets a copy of the texture data.
Texture3D.GetData (T[], Int32, Int32)	Gets a copy of the texture data, specifying the starting index and number of elements to copy.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D.GetData Generic Method (Int32, Int32, Int32, Int32, Int32, Int32, Int32, T[], Int32, Int32)

Gets a copy of the texture data, specifying the level and dimensions of the volume texture to copy.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (  
    int level,  
    int left,  
    int top,  
    int right,  
    int bottom,  
    int front,  
    int back,  
    T[] data,  
    int startIndex,  
    int elementCount  
) where T : ValueType
```

Type Parameters

T

Type of value to return.

Parameters

level

The mipmap level where the data will be placed.

left

Position of the left side of the box on the x-axis.

top

Position of the top of the box on the y-axis.

right

Position of the right side of the box on the x-axis.

bottom

Position of the bottom of the box on the y-axis.

front

Position of the front of the box on the z-axis.

back

Position of the back of the box on the z-axis.

data

An array to fill with data.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.GetData Generic Method (T[])

Gets a copy of the texture data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (  
    T[] data  
) where T : ValueType
```

Type Parameters

T

Type of value to return.

Parameters

data

An array to fill with data.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.GetData Generic Method (T[], Int32, Int32)

Gets a copy of the texture data, specifying the starting index and number of elements to copy.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (  
    T[] data,  
    int startIndex,  
    int elementCount  
) where T : ValueType
```

Type Parameters

T

Type of value to return.

Parameters

data

An array to fill with data.

startIndex

Index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.SetData Method

Copies array data to the texture.

Overload List

Name	Description
Texture3D.SetData (Int32, Int32, Int32, Int32, Int32, Int32, Int32, T[], Int32, Int32, SetDataOptions)	Copies array data to the texture, specifying the dimensions of the volume and the mipmap level where the data is to be placed.
Texture3D.SetData (T[])	Copies array data to the texture at mipmap level 0.
Texture3D.SetData (T[], Int32, Int32, SetDataOptions)	Copies array data to the texture at mipmap level 0, specifying the starting index and number of elements to copy.

Exceptions

Exception type	Condition
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D.SetData Generic Method (Int32, Int32, Int32, Int32, Int32, Int32, Int32, T[], Int32, Int32, SetDataOptions)

Copies array data to the texture, specifying the dimensions of the volume and the mipmap level where the data is to be placed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    int level,
    int left,
    int top,
    int right,
    int bottom,
    int front,
    int back,
    T[] data,
    int startIndex,
    int elementCount,
    SetDataOptions options
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

level

The mipmap level where the data will be placed.

left

Position of the left side of the box on the x-axis.

top

Position of the top of the box on the y-axis.

right

Position of the right side of the box on the x-axis.

bottom

Position of the bottom of the box on the y-axis.

front

Position of the front of the box on the z-axis.

back

Position of the back of the box on the z-axis.

data

The array of data to copy. The number of elements in the array should be equal to the size of the box where the data will be placed.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy.

options

Option specifying whether existing data in the buffer will be kept after this operation.

Exceptions

Exception type	Condition
ArgumentException	<i>data</i> is null .
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.SetData Generic Method (T[])

Copies array data to the texture at mipmap level 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    T[] data
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy. The number of elements in the array must be equal to the size of the texture, which is **Width** × **Height** × **Depth**.

Exceptions

Exception type	Condition
ArgumentException	<i>data</i> is null .
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.SetData Generic Method (T[], Int32, Int32, SetDataOptions)

Copies array data to the texture at mipmap level 0, specifying the starting index and number of elements to copy.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    T[] data,
    int startIndex,
    int elementCount,
    SetDataOptions options
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

data

The array of data to copy.

startIndex

The index of the element in the array at which to start copying.

elementCount

The number of elements to copy. This must be equal to the size of the texture, which is [Width](#) × [Height](#) × [Depth](#).

options

Option specifying if existing data in the buffer will be kept after this operation.

Exceptions

Exception type	Condition
ArgumentException	<i>data</i> is null .
InvalidOperationException	The size of the data passed in is too large or too small for this resource.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[Texture3D Class](#)














[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D Properties

Public Properties

	Name	Description
	Depth	Gets the depth of this volume texture resource, in pixels.
	Format	Gets the pixel format for this texture resource.
	GraphicsDevice	(Inherited from GraphicsResource .)
	Height	Gets the height of this texture resource, in pixels.
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	(Inherited from Texture .)
	LevelOfDetail	(Inherited from Texture .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Tag	(Inherited from GraphicsResource .)
	TextureUsage	Gets the state of the related TextureUsage enumeration.
	Width	Gets the width of this texture resource, in pixels.

See Also

Reference

[Texture3D Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Texture3D.Depth Property

Gets the depth of this volume texture resource, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Depth { get; }
```

Property Value

The depth of this volume texture resource, in pixels.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.Format Property

Gets the pixel format for this texture resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; }
```

Property Value

The pixel format of this texture resource.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.Height Property

Gets the height of this texture resource, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

The height of this texture resource, in pixels.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.TextureUsage Property

Gets the state of the related [TextureUsage](#) enumeration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureUsage TextureUsage { get; }
```

Property Value

Indicates how the application uses the related texture.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Texture3D.Width Property

Gets the width of this texture resource, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

The width of this texture resource, in pixels.

See Also

Reference

[Texture3D Class](#)

[Texture3D Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureAddressMode Enumeration

Defines constants that describe supported texture-addressing modes.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum TextureAddressMode
```

Members

Member name	Description
Border	Texture coordinates outside the range [0.0, 1.0] are set to the border color.
Clamp	Texture coordinates outside the range [0.0, 1.0] are set to the texture color at 0.0 or 1.0, respectively.
Mirror	Similar to Wrap , except that the texture is flipped at every integer junction. For u values between 0 and 1, for example, the texture is addressed normally; between 1 and 2, the texture is flipped (mirrored); between 2 and 3, the texture is normal again, and so on.
MirrorOnce	Similar to Mirror and Clamp . Takes the absolute value of the texture coordinate (thus, mirroring around 0), and then clamps to the maximum value. The most common usage is for volume textures, where support for the full MirrorOnce texture-addressing mode is not necessary, but the data is symmetrical around the one axis.
Wrap	Tile the texture at every integer junction. For example, for u values between 0 and 3, the texture is repeated three times; no mirroring is performed.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[AddressU](#)

[AddressV](#)

[AddressW](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCollection Class

Represents a collection of [Texture](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class TextureCollection
```

See Also

Reference

[TextureCollection Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






TextureCollection Members

The following tables list the members exposed by the TextureCollection type.



Public Properties

	Name	Description
	Item	Gets or sets the Texture at the specified sampler number.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[TextureCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[TextureCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCollection Properties

Public Properties

	Name	Description
	Item	Gets or sets the Texture at the specified sampler number.

See Also

Reference

[TextureCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCollection.Item Property

Gets or sets the [Texture](#) at the specified sampler number.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture this [
    int index
] { get; set; }
```

Property Value

The [Texture](#) at the specified index.

Exceptions

Exception type	Condition
InvalidOperationException	The sampler number is invalid. It is either less than zero or greater than the maximum number of textures available for the graphics device.
ObjectDisposedException	Item was called after the GraphicsDevice was disposed.

See Also

Reference

[GraphicsDevice.SamplerStates](#) Property

[GraphicsDevice.Textures](#) Property

[GraphicsDevice.VertexTextures](#) Property

[TextureCollection](#) Class

[TextureCollection](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCreationParameters Structure

Note

This structure is available only when developing for Windows.

Describes the parameters to use when initializing a new instance of a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct TextureCreationParameters
```

See Also

Reference

[TextureCreationParameters Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista











TextureCreationParameters Members

The following tables list the members exposed by the TextureCreationParameters type.








Public Constructors

	Name	Description
	TextureCreationParameters	Creates a new instance of TextureCreationParameters .



Public Properties

	Name	Description
	ColorKey	Gets or sets the color value to replace with transparent black.
	Default	Gets the default texture creation parameters.
	Depth	Gets or sets the depth to create the Texture with.
	Filter	Gets or sets a set of options controlling how the image is filtered.
	Format	Gets or sets the format of the texture to be created.
	Height	Gets or sets the height to create a Texture with.
	MipFilter	Gets or sets a set of options controlling how mipmaps are filtered.
	MipLevels	Gets or sets the number of mip levels to create a Texture with.
	TextureUsage	Gets the state of the related TextureUsage enumeration.
	Width	Gets or sets the width to create a Texture with.

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of TextureCreationParameters are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two instances of TextureCreationParameters are equal.
	op_Inequality	Determines whether two instances of TextureCreationParameters are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current TextureCreationParameters .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureCreationParameters Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCreationParameters Constructor

Note

This constructor is available only when developing for Windows.

Creates a new instance of [TextureCreationParameters](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCreationParameters (
    int width,
    int height,
    int depth,
    int mipLevels,
    SurfaceFormat format,
    TextureUsage textureUsage,
    Color colorKey,
    FilterOptions filter,
    FilterOptions mipFilter
)
```

Parameters

width

Width of the texture.

height

Height of the texture.

depth

Depth of the texture.

mipLevels

Number of mip levels in the texture.

format

The format of the texture.

textureUsage

A set of options identifying the behaviors of this texture resource.

colorKey

The color key of the texture.

filter

A set of options controlling how the image is filtered.

mipFilter

A set of options controlling how mipmaps are filtered.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset Event](#)

[TextureCreationParameters Structure](#)








[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of TextureCreationParameters are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two instances of TextureCreationParameters are equal.
	Op_Inequality	Determines whether two instances of TextureCreationParameters are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String that represents the current TextureCreationParameters .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureCreationParameters Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCreationParameters.Equals Method

Determines whether two instances of [TextureCreationParameters](#) are equal.

Overload List

Name	Description
TextureCreationParameters.Equals (Object)	Determines whether the specified Object is equal to the TextureCreationParameters .
TextureCreationParameters.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCreationParameters.Equals Method (Object)

Note

This method is available only when developing for Windows.

Determines whether the specified [Object](#) is equal to the [TextureCreationParameters](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [TextureCreationParameters](#).

Return Value

true if the specified [Object](#) is equal to the current [TextureCreationParameters](#); **false** otherwise.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCreationParameters.GetHashCode Method

Note

This method is available only when developing for Windows.

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [TextureCreationParameters](#).

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two instances of [TextureCreationParameters](#) are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    TextureCreationParameters left,  
    TextureCreationParameters right  
)
```

Parameters

left

The object to the left of the equality operator.

right

The object to the right of the equality operator.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCreationParameters.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two instances of [TextureCreationParameters](#) are not equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    TextureCreationParameters left,  
    TextureCreationParameters right  
)
```

Parameters

left

The object to the left of the inequality operator.

right

The object to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCreationParameters.ToString Method

Note

This method is available only when developing for Windows.

Returns a [String](#) that represents the current [TextureCreationParameters](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of the current [TextureCreationParameters](#).

See Also

Reference

[TextureCreationParameters Structure](#)











[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters Properties

Public Properties

	Name	Description
	ColorKey	Gets or sets the color value to replace with transparent black.
 S	Default	Gets the default texture creation parameters.
	Depth	Gets or sets the depth to create the Texture with.
	Filter	Gets or sets a set of options controlling how the image is filtered.
	Format	Gets or sets the format of the texture to be created.
	Height	Gets or sets the height to create a Texture with.
	MipFilter	Gets or sets a set of options controlling how mipmaps are filtered.
	MipLevels	Gets or sets the number of mip levels to create a Texture with.
	TextureUsage	Gets the state of the related TextureUsage enumeration.
	Width	Gets or sets the width to create a Texture with.

See Also

Reference

[TextureCreationParameters Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCreationParameters.ColorKey Property

Note

This property is available only when developing for Windows.

Gets or sets the color value to replace with transparent black.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color ColorKey { get; set; }
```

Property Value

Color value to replace with transparent black.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.Default Property

Note

This property is available only when developing for Windows.

Gets the default texture creation parameters.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCreationParameters Default { get; }
```

Property Value

The default parameters.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.Depth Property

Note

This property is available only when developing for Windows.

Gets or sets the depth to create the [Texture](#) with.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Depth { get; set; }
```

Property Value

The depth.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.Filter Property

Note

This property is available only when developing for Windows.

Gets or sets a set of options controlling how the image is filtered.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FilterOptions Filter { get; set; }
```

Property Value

A set of options controlling how the image is filtered.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.Format Property

Note

This property is available only when developing for Windows.

Gets or sets the format of the texture to be created.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; set; }
```

Property Value

The format of the texture to be created.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.Height Property

Note

This property is available only when developing for Windows.

Gets or sets the height to create a [Texture](#) with.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; set; }
```

Property Value

The height.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.MipFilter Property

Note

This property is available only when developing for Windows.

Gets or sets a set of options controlling how mipmaps are filtered.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public FilterOptions MipFilter { get; set; }
```

Property Value

A set of options controlling how mipmaps are filtered.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.MipLevels Property

Note

This property is available only when developing for Windows.

Gets or sets the number of mip levels to create a [Texture](#) with.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MipLevels { get; set; }
```

Property Value

The number of mip levels.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.TextureUsage Property

Note

This property is available only when developing for Windows.

Gets the state of the related [TextureUsage](#) enumeration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureUsage TextureUsage { get; set; }
```

Property Value

Indicates how the application uses the related texture.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCreationParameters.Width Property

Note

This property is available only when developing for Windows.

Gets or sets the width to create a [Texture](#) with.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; set; }
```

Property Value

The width.

See Also

Reference

[TextureCreationParameters Structure](#)

[TextureCreationParameters Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCube Class

Represents a set of six 2D textures, one for each face of a cube.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class TextureCube : Texture
```

Remarks

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A cube texture is a collection of six textures, one for each face of the cube. All faces must be present in the cube texture. Also, a cube map surface must be the same pixel size in all three dimensions (x, y, and z). Figure 1 shows a fully populated texture cube.

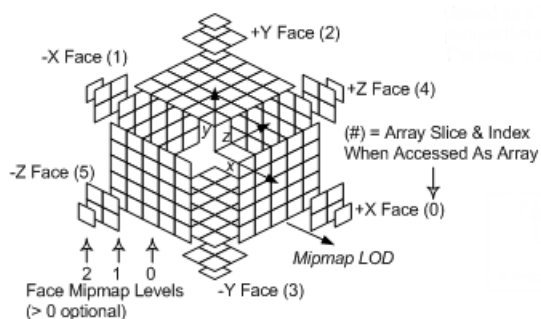


Figure 1. TextureCube Resource Architecture

See Also

Tasks

[How To: Create a SkySphere](#)

Reference

[TextureCube Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista












TextureCube Members

The following tables list the members exposed by the TextureCube type.















Public Constructors

Name	Description
 TextureCube	Creates an uninitialized TextureCube resource of the specified dimensions. To initialize a Texture2D from an existing file, see the static method ContentManager.Load or FromFile .




Public Properties

Name	Description
 Format	Gets the pixel format for this texture resource.
 GraphicsDevice	(Inherited from GraphicsResource .)
 IsDisposed	(Inherited from GraphicsResource .)
 LevelCount	(Inherited from Texture .)
 LevelOfDetail	(Inherited from Texture .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 Size	Gets the width and height of this texture resource, in pixels.
 Tag	(Inherited from GraphicsResource .)
 TextureUsage	Gets the state of the related TextureUsage enumeration.


Public Methods

Name	Description
 Dispose	Overloaded. Releases the resources used by the TextureCube class.
 Equals	(Inherited from Object .)
  FromFile	Overloaded. Creates a cube texture resource from a file.
 GenerateMipMaps	(Inherited from Texture .)
 GetCreationParameters	(Inherited from Texture .)
 GetData	Overloaded. Returns a copy of the texture data.
 GetHashCode	(Inherited from Object .)
 GetTextureInformation	(Inherited from Texture .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Save	(Inherited from Texture .)
 SetData	Overloaded. Copies array data to the texture.
 ToString	Returns a String that represents the current TextureCube .

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[TextureCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube Constructor

Creates an uninitialized [TextureCube](#) resource of the specified dimensions. To initialize a [Texture2D](#) from an existing file, see the static method [ContentManager.Load](#) or [FromFile](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureCube (
    GraphicsDevice graphicsDevice,
    int size,
    int numberLevels,
    TextureUsage usage,
    SurfaceFormat format
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) that will display the cube texture.

size

The size of the edges of all the top-level faces of the cube texture. The pixel dimensions of subsequent levels of each face will be the truncated value of half of the previous level's pixel dimension (independently). Each dimension clamps at a size of 1 pixel. Thus, if the division by 2 results in 0 (zero), 1 will be taken instead.

numberLevels

The number of downsampled surfaces to create for each face of the cube texture when preprocessing the texture. These smaller versions of the texture, known as mip levels, are used when the texture is minified to fit a smaller area than the original texture size. The chain of downsampled surfaces associated with a texture is sometimes called a mipmap chain.

If *numberLevels* is zero, all cube texture sublevels down to 1×1 pixels will be generated for each face for hardware that supports mipmapped cube textures. Use [LevelCount](#) to see the number of levels generated.

usage

A set of options identifying the behaviors of this resource.

format

The format of all levels in the cube texture.

Exceptions

Exception type	Condition
ArgumentException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> The <i>size</i> requested would result in a resource too large for the graphics device. <i>numberLevels</i> is too large. It is either greater than 32, or if less than 32 it would result in a cube texture that is too large for this graphics device. Textures have a maximum of 32 levels. <i>usage</i> is not a valid value. Texture resources can only be created with the following TextureUsage options: TextureUsage.AutoGenerateMipMap, TextureUsage.Tiled, and TextureUsage.Linear. <i>format</i> is SurfaceFormat.Unknown. Textures cannot be created using SurfaceFormat.Unknown. <p>The parameters specified are not compatible:</p> <ul style="list-style-type: none"> <i>format</i> and <i>numberLevels</i> are incompatible. Multi-element textures of type SurfaceFormat.Multi2Bgra32 cannot be mipmapped, set <i>numberLevels</i> to 1. <i>usage</i> and <i>numberLevels</i> are incompatible. Textures specifying the usage TextureUsage.AutoGenerateMipMap must set <i>numberLevels</i> to either 0 or 1.
ArgumentNullException	<i>graphicsDevice</i> is null .
ArgumentOutOfRangeException	<i>size</i> is less than or equal to zero. <i>size</i> must be greater than or equal to zero.

Remarks

A mipmap (texture) is a collection of successively downsampled (mipmapped) surfaces. An application can discover support for automatic generation of mipmaps in a particular format by calling [CheckDeviceFormat](#) with [TextureUsage.AutoGenerateMipMap](#).

A texel represents the smallest unit of a texture that can be read from or written to by the GPU. A texel is composed of 1 to 4 components. Specifically, a texel may be any one of the available texture formats represented in the [SurfaceFormat](#) enumeration.

A cube texture is a collection of six textures, one for each face of the cube. All faces must be present in the cube texture. Also, a cube map surface must be the same pixel size in all three dimensions (x, y, and z). Figure 1 shows a fully populated texture cube.

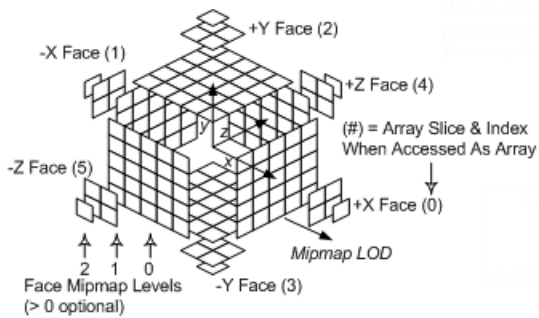


Figure 1. TextureCube Resource Architecture

See Also

Reference

[ContentManager.Load Generic Method](#)

[TextureCube.FromFile Method](#)

[TextureCube Class](#)















[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Releases the resources used by the TextureCube class.
	Equals	(Inherited from Object .)
 	FromFile	Overloaded. Creates a cube texture resource from a file.
	GenerateMipMaps	(Inherited from Texture .)
	GetCreationParameters	(Inherited from Texture .)
	GetData	Overloaded. Returns a copy of the texture data.
	GetHashCode	(Inherited from Object .)
	GetTextureInformation	(Inherited from Texture .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Save	(Inherited from Texture .)
	SetData	Overloaded. Copies array data to the texture.
	ToString	Returns a String that represents the current TextureCube .

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[TextureCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube.Dispose Method

Releases the resources used by the [TextureCube](#) class.

Overload List

Name	Description
TextureCube.Dispose (Boolean)	Releases the unmanaged resources used by the TextureCube and optionally releases the managed resources.
TextureCube.Dispose ()	(Inherited from GraphicsResource .)

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube.Dispose Method (Boolean)

Releases the unmanaged resources used by the [TextureCube](#) and optionally releases the managed resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [TextureCube](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.FromFile Method

Creates a cube texture resource from a file.

Overload List

Name	Description
TextureCube.FromFile (GraphicsDevice, Stream)	Creates a cube texture resource from a stream.
TextureCube.FromFile (GraphicsDevice, Stream, Int32)	Creates a texture resource from a stream, specifying the number of bytes in the stream.
TextureCube.FromFile (GraphicsDevice, Stream, Int32, TextureCreationParameters)	Creates a texture resource from a stream, specifying the number of bytes in the stream and the texture creation parameters to use.
TextureCube.FromFile (GraphicsDevice, Stream, TextureCreationParameters)	Creates a texture resource from a stream, specifying the parameters to be used in creation of the texture.
TextureCube.FromFile (GraphicsDevice, String)	Creates a cube texture resource from a file.
TextureCube.FromFile (GraphicsDevice, String, Int32)	Creates a cube texture resource from a file, specifying the width and height of the texture in pixels.
TextureCube.FromFile (GraphicsDevice, String, TextureCreationParameters)	Creates a texture resource from a file, specifying the parameters to be used in creation of the texture.
TextureCube.FromFile (GraphicsDevice, String, Int32, Int32, Int32)	(Inherited from Texture .)

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube.FromFile Method (GraphicsDevice, Stream)

Note

This method is available only when developing for Windows.

Creates a cube texture resource from a stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream  
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

Return Value

The cube texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCube.FromFile Method (GraphicsDevice, Stream, Int32)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the number of bytes in the stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (  
    GraphicsDevice graphicsDevice,  
    Stream textureStream,  
    int numberBytes  
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

numberBytes

The number of bytes in *textureStream*.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCube.FromFile Method (GraphicsDevice, Stream, Int32, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the number of bytes in the stream and the texture creation parameters to use.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    int numberBytes,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the volume texture data.

numberBytes

The number of bytes in *textureStream*.

creationParameters

The parameters to use when creating this texture.

Return Value

The volume texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCube.FromFile Method (GraphicsDevice, Stream, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a stream, specifying the parameters to be used in creation of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (
    GraphicsDevice graphicsDevice,
    Stream textureStream,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

textureStream

Stream containing the texture data.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentNullException	<i>textureStream</i> is null . The stream passed into this method cannot be null.
NotSupportedException	<i>textureStream</i> is not readable. Streams passed into this method must be readable.
EndOfStreamException	<i>textureStream</i> does not contain enough data to support this call.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCube.FromFile Method (GraphicsDevice, String)

Note

This method is available only when developing for Windows.

Creates a cube texture resource from a file.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (
    GraphicsDevice graphicsDevice,
    string filename
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

Return Value

The texture resource that has been created on the specified graphics device.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCube.FromFile Method (GraphicsDevice, String, Int32)

Note

This method is available only when developing for Windows.

Creates a cube texture resource from a file, specifying the width and height of the texture in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    int size
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

size

Width and height of the cube texture, in pixels.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCube.FromFile Method (GraphicsDevice, String, TextureCreationParameters)

Note

This method is available only when developing for Windows.

Creates a texture resource from a file, specifying the parameters to be used in creation of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TextureCube FromFile (
    GraphicsDevice graphicsDevice,
    string filename,
    TextureCreationParameters creationParameters
)
```

Parameters

graphicsDevice

The graphics device to use to create the texture resource.

filename

The name of the file containing the texture.

creationParameters

The parameters to use when creating this texture.

Return Value

The texture resource that has been created on the specified graphics device.

Exceptions

Exception type	Condition
ArgumentException	<i>filename</i> is a zero-length string, contains only white space, or contains one or more invalid characters as defined by InvalidPathChars .
ArgumentNullException	<i>filename</i> is null .
PathTooLongException	The specified path, file name, or both exceed the system-defined maximum length. For example, on Windows-based platforms, paths must be less than 248 characters, and file names must be less than 260 characters.
DirectoryNotFoundException	The specified path is invalid (for example, it is on an unmapped drive).
UnauthorizedAccessException	The <i>filename</i> parameter specifies a directory, or the caller does not have the required permission to access the file specified by <i>filename</i> .
FileNotFoundException	The file specified in <i>filename</i> was not found.
NotSupportedException	<i>filename</i> is in an invalid format.

Remarks

The supported texture image file formats are described by the [ImageFileFormat](#) enumeration.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCube.GetData Method

Returns a copy of the texture data.

Overload List

Name	Description
TextureCube.GetData (CubeMapFace, Int32, Nullable<Rectangle>, T[], Int32, Int32)	Returns a copy of the texture data, specifying the start index, starting offset, number of elements, region to copy, and level where the data is to be placed.
TextureCube.GetData (CubeMapFace, T[])	Returns a copy of the texture data.
TextureCube.GetData (CubeMapFace, T[], Int32, Int32)	Returns a copy of the texture data, specifying the start index and number of elements in the vertex buffer.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube.GetData Generic Method (CubeMapFace, Int32, Nullable<Rectangle>, T[], Int32, Int32)

Returns a copy of the texture data, specifying the start index, starting offset, number of elements, region to copy, and level where the data is to be placed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    CubeMapFace faceType,
    int level,
    Nullable<Rectangle> rect,
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of data in the texture buffer.

Parameters

faceType

The cube map face type.

level

The mipmap level where the data will be placed.

rect

The section of the texture where the data will be placed. **null** indicates the data will be copied over the entire texture.

data

The array into which to copy the data.

startIndex

Index in the array at which to begin the copy.

elementCount

Number of elements in the array.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> must be of sufficient length to receive the data.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.GetData Generic Method (CubeMapFace, T[])

Returns a copy of the texture data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (  
    CubeMapFace faceType,  
    T[] data  
) where T : ValueType
```

Type Parameters

T

The type of data in the texture buffer.

Parameters

faceType

The cube map face type.

data

The array into which to copy the data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> must be of sufficient length to receive the data.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.GetData Generic Method (CubeMapFace, T[], Int32, Int32)

Returns a copy of the texture data, specifying the start index and number of elements in the vertex buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (  
    CubeMapFace faceType,  
    T[] data,  
    int startIndex,  
    int elementCount  
) where T : ValueType
```

Type Parameters

T

The type of data in the texture buffer.

Parameters

faceType

The cube map face type.

data

The array into which to copy the data.

startIndex

Index in the array at which to begin the copy.

elementCount

Number of elements in the array.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> must be of sufficient length to receive the data.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.SetData Method

Copies array data to the texture.

Overload List

Name	Description
TextureCube.SetData (CubeMapFace, Int32, Nullable<Rectangle>, T[], Int32, Int32, SetDataOptions)	Copies array data to the texture, specifying a start offset, mip map level, and subregion to copy.
TextureCube.SetData (CubeMapFace, T[])	Copies array data to the texture at mipmap level 0.
TextureCube.SetData (CubeMapFace, T[], Int32, Int32, SetDataOptions)	Copies array data to the texture at mipmap level 0, specifying a start offset.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube.SetData Generic Method (CubeMapFace, Int32, Nullable<Rectangle>, T[], Int32, Int32, SetDataOptions)

Copies array data to the texture, specifying a start offset, mipmap level, and subregion to copy.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    CubeMapFace faceType,
    int level,
    Nullable<Rectangle> rect,
    T[] data,
    int startIndex,
    int elementCount,
    SetDataOptions options
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

faceType

The cube map face type.

level

The mipmap level where the data will be placed.

rect

The section of the texture where the data will be placed. **null** indicates the data will be copied over the entire texture.

data

The array of data to copy.

startIndex

Start offset in the array.

elementCount

Number of elements in the array. The number of elements to copy must be equal to the size of the texture.

options

Option specifying if existing data in the buffer will be kept after this operation.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.SetData Generic Method (CubeMapFace, T[])

Copies array data to the texture at mipmap level 0.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (  
    CubeMapFace faceType,  
    T[] data  
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

faceType

The cube map face type.

data

The array of data to copy. The number of elements in the array must be equal to the size of the texture.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.SetData Generic Method (CubeMapFace, T[], Int32, Int32, SetDataOptions)

Copies array data to the texture at mipmap level 0, specifying a start offset.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    CubeMapFace faceType,
    T[] data,
    int startIndex,
    int elementCount,
    SetDataOptions options
) where T : ValueType
```

Type Parameters

T

The type of the elements in the array.

Parameters

faceType

The cube map face type.

data

The array of data to copy.

startIndex

Start offset in the array.

elementCount

Number of elements in the array. This must be equal to the size of the texture.

options

Option specifying if existing data in the buffer will be kept after this operation.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes.

This is because, in cases where the size of the back buffer and depth-stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

See Also

Concepts

[Predicated Tiling](#)

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.ToString Method

Returns a [String](#) that represents the current [TextureCube](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

A [String](#) that represents the current [TextureCube](#).

See Also

Reference

[TextureCube Class](#)












[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube Properties

Public Properties

	Name	Description
	Format	Gets the pixel format for this texture resource.
	GraphicsDevice	(Inherited from GraphicsResource .)
	IsDisposed	(Inherited from GraphicsResource .)
	LevelCount	(Inherited from Texture .)
	LevelOfDetail	(Inherited from Texture .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	Size	Gets the width and height of this texture resource, in pixels.
	Tag	(Inherited from GraphicsResource .)
	TextureUsage	Gets the state of the related TextureUsage enumeration.

See Also

Reference

[TextureCube Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureCube.Format Property

Gets the pixel format for this texture resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; }
```

Property Value

The pixel format of this texture resource.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.Size Property

Gets the width and height of this texture resource, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Size { get; }
```

Property Value

The size of this texture resource, in pixels.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureCube.TextureUsage Property

Gets the state of the related [TextureUsage](#) enumeration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureUsage TextureUsage { get; }
```

Property Value

Indicates how the application uses the related texture.

See Also

Reference

[TextureCube Class](#)

[TextureCube Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureFilter Enumeration

Defines how a texture will be filtered as it is minified for each mipmap level.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum TextureFilter
```

Members

Member name	Description
Anisotropic	Anisotropic texture filtering used as a texture magnification or minification filter. This type of filtering compensates for distortion caused by the difference in angle between the texture polygon and the plane of the screen.
GaussianQuad	A 4-sample Gaussian filter used as a texture magnification or minification filter.
Linear	Bilinear interpolation filtering used as a texture magnification or minification filter. A weighted average of a 2×2 area of texels surrounding the desired pixel is used. The texture filter used between mipmap levels is trilinear mipmap interpolation, in which the rasterizer performs linear interpolation on pixel color, using the texels of the two nearest mipmap textures.
None	Mipmapping disabled. The rasterizer uses the magnification filter instead.
Point	Point filtering used as a texture magnification or minification filter. The texel with coordinates nearest to the desired pixel value is used. The texture filter used between mipmap levels is based on the nearest point; that is, the rasterizer uses the color from the texel of the nearest mipmap texture.
PyramidalQuad	A 4-sample tent filter used as a texture magnification or minification filter.

Remarks

To determine whether a format supports texture filter types other than **Point** (which is always supported), call [CheckDeviceFormat](#).

Not all valid filtering modes for a device apply to volume maps. In general, **Point** and **Linear** magnification filters are supported for volume maps. If [SupportsMipVolumeMap](#) is **true** (the texture capabilities can be found in [TextureCapabilities](#)), the **Point** mipmap filter and the **Point** and **Linear** minification filters are supported for volume maps. The device might or might not support the **Linear** mipmap filter for volume maps. Devices that support anisotropic filtering for 2D maps do not necessarily support it for volume maps. However, applications that enable anisotropic filtering, even if they do not support it, receive the best available filtering (probably linear).

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[GraphicsDeviceCapabilities.FilterCaps](#) Properties

[SamplerState.MagFilter](#) Property

[SamplerState.MinFilter](#) Property

[SamplerState.MipFilter](#) Property

[Texture.GenerateMipMaps](#) Method

[QueryUsages](#) Enumeration

[Microsoft.Xna.Framework.Graphics](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista

TextureInformation Structure

Note

This structure is available only when developing for Windows.

Encapsulates information describing texture resources.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct TextureInformation
```

See Also

Reference

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista








TextureInformation Members

The following tables list the members exposed by the TextureInformation type.








Public Constructors

Name	Description
 TextureInformation	Initializes a new instance of the TextureInformation class.



Public Properties

Name	Description
 Depth	Gets or sets the depth of a texture.
 Format	Gets or sets the format of the texture.
 Height	Gets or sets the height, in pixels, of a texture.
 ImageFormat	Gets the file format of the original image.
 MipLevels	Gets or sets the number of mip levels of a texture.
 ResourceType	Gets the type of this resource.
 Width	Gets or sets the width, in pixels, of a texture.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether two instances of TextureInformation are equal.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two instances of TextureInformation are equal.
 Op_Inequality	Determines whether two instances of TextureInformation are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureInformation Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureInformation Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [TextureInformation](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TextureInformation (  
    int width,  
    int height,  
    int depth,  
    int mipLevels,  
    SurfaceFormat format  
)
```

Parameters

width

Width of the texture, in pixels.

height

Height of the texture, in pixels.

depth

Depth of the texture, in pixels.

mipLevels

Number of mip levels in original image.

format

Describes the data in the original image.

See Also

Reference

[TextureInformation Structure](#)








[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

TextureInformation Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether two instances of TextureInformation are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two instances of TextureInformation are equal.
	op_Inequality	Determines whether two instances of TextureInformation are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureInformation Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureInformation.Equals Method

Determines whether two instances of [TextureInformation](#) are equal.

Overload List

Name	Description
TextureInformation.Equals (Object)	Determines whether the specified Object is equal to the TextureInformation .
TextureInformation.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureInformation.Equals Method (Object)

Note

This method is available only when developing for Windows.

Determines whether the specified [Object](#) is equal to the [TextureInformation](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [TextureInformation](#).

Return Value

true if the specified [Object](#) is equal to the current [TextureInformation](#); **false** otherwise.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureInformation.GetHashCode Method

Note

This method is available only when developing for Windows.

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

A hash code for the current [TextureInformation](#).

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.op_Equality Method

Note

This method is available only when developing for Windows.

Determines whether two instances of [TextureInformation](#) are equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    TextureInformation left,  
    TextureInformation right  
)
```

Parameters

left

The object to the left of the equality operator.

right

The object to the right of the equality operator.

Return Value

true if *left* is equal to *right*; **false** otherwise.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureInformation.op_Inequality Method

Note

This method is available only when developing for Windows.

Determines whether two instances of [TextureInformation](#) are not equal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    TextureInformation left,  
    TextureInformation right  
)
```

Parameters

left

The object to the left of the inequality operator.

right

The object to the right of the inequality operator.

Return Value

true if *left* is not equal to *right*; **false** otherwise.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureInformation.ToString Method

Note

This method is available only when developing for Windows.

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[TextureInformation Structure](#)








[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation Properties

Public Properties

	Name	Description
	Depth	Gets or sets the depth of a texture.
	Format	Gets or sets the format of the texture.
	Height	Gets or sets the height, in pixels, of a texture.
	ImageFormat	Gets the file format of the original image.
	MipLevels	Gets or sets the number of mip levels of a texture.
	ResourceType	Gets the type of this resource.
	Width	Gets or sets the width, in pixels, of a texture.

See Also

Reference

[TextureInformation Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

TextureInformation.Depth Property

Note

This property is available only when developing for Windows.

Gets or sets the depth of a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Depth { get; set; }
```

Property Value

The depth of the texture.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.Format Property

Note

This property is available only when developing for Windows.

Gets or sets the format of the texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SurfaceFormat Format { get; set; }
```

Property Value

The format of the texture.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.Height Property

Note

This property is available only when developing for Windows.

Gets or sets the height, in pixels, of a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; set; }
```

Property Value

The height, in pixels, of the texture.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.ImageFormat Property

Note

This property is available only when developing for Windows.

Gets the file format of the original image.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ImageFileFormat ImageFormat { get; }
```

Property Value

The file format of the original image.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.MipLevels Property

Note

This property is available only when developing for Windows.

Gets or sets the number of mip levels of a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MipLevels { get; set; }
```

Property Value

The number of mip levels of the texture.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.ResourceType Property

Note

This property is available only when developing for Windows.

Gets the type of this resource.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ResourceType ResourceType { get; }
```

Property Value

The resource type, identifying this resource as a texture.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureInformation.Width Property

Note

This property is available only when developing for Windows.

Gets or sets the width, in pixels, of a texture.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; set; }
```

Property Value

The width, in pixels, of the texture.

See Also

Reference

[TextureInformation Structure](#)

[TextureInformation Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureUsage Enumeration

Specifies special usage of the texture data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum TextureUsage
```

Members

Member name	Description
AutoGenerateMipMap	The resource automatically generates mipmaps. Automatic generation of mipmaps is not supported for volume textures and depth stencil surfaces/textures.
Linear	(Xbox 360 only) Linear texture memory format. Linear formats are arranged in memory using a left-to-right and top-to-bottom layout (with pixels contiguous in memory for each row, and rows contiguous except for possible alignment padding). Use this format when doing dynamic CPU updates to textures where: <ul style="list-style-type: none"> You do not want the overhead of translating tile addresses using the CPU. Textures are not often referenced during the frame. Less efficient GPU performance is not a problem. <p>This flag is valid only for texture creation on Xbox 360. It is ignored if used elsewhere. On Xbox, most formats are created as Tiled if it is applicable.</p>
None	None.
Tiled	(Xbox 360 only) Tiled texture memory format. Tiled formats are arranged using a memory layout designed to maximize the cache coherency of texture fetches and balance the access workload of the memory controllers. The exact pixel arrangement of tiled formats varies based on the bit depth of the format; however, pixels that are close to one another spatially are located close to one another in memory. For example, two pixels adjacent to one another in a vertical column would be separated in memory by an entire row of pixels for a linear format, but could be separated by only a few bytes for a tiled format. <p>The tiled texture format is usually more efficient than the linear texture format.</p> <p>This flag is valid only when used for texture creation on Xbox 360, and will be ignored if used elsewhere. On Xbox, most formats are created as Tiled if it is applicable.</p>

See Also

Reference

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

TextureWrapCoordinates Enumeration

Defines supported wrap coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum TextureWrapCoordinates
```

Members

Member name	Description
None	No texture wrap coordinates specified.
Zero	U texture wrapping (wrapping in the direction of the first dimension).
One	V texture wrapping (wrapping in the direction of the second dimension).
Two	W texture wrapping (wrapping in the direction of the third dimension).
Three	Texture wrapping in the direction of the fourth dimension.

RemarksThis enumeration is used by the [Wrap0](#) property and the [Wrap1](#) through [Wrap15](#) properties.

See Also

Concepts

[What Is Texture Mapping?](#)

Reference

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

VertexBuffer Class

Represents a list of 3D vertices to be streamed to the graphics device.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class VertexBuffer : GraphicsResource
```

Example

The vertex stream of the graphics device must be set before any call to [DrawPrimitives](#). The following example associates a user created vertex buffer of type **VertexPositionNormalTexture** with vertex stream 0 (zero) of the graphics device.

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);
```

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[VertexBuffer Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista









VertexBuffer Members

The following tables list the members exposed by the VertexBuffer type.









Public Constructors

Name	Description
 VertexBuffer	Overloaded. Initializes a new instance of the VertexBuffer class.



Public Properties

Name	Description
 BufferUsage	Gets the state of the related BufferUsage enumeration.
 GraphicsDevice	(Inherited from GraphicsResource .)
 IsDisposed	(Inherited from GraphicsResource .)
 Name	(Inherited from GraphicsResource .)
 Priority	(Inherited from GraphicsResource .)
 ResourceType	(Inherited from GraphicsResource .)
 SizeInBytes	Gets the size, in bytes, of this vertex buffer.
 Tag	(Inherited from GraphicsResource .)


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetData	Overloaded. Returns a copy of the vertex buffer data.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetData	Overloaded. Sets the vertex buffer data.
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	(Inherited from GraphicsResource .)

Public Events

Name	Description
 Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[VertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer Constructor

Initializes a new instance of the [VertexBuffer](#) class.

Overload List

Name	Description
VertexBuffer (GraphicsDevice, Int32, BufferUsage)	Initializes a new instance of VertexBuffer , specifying the size and usage.
VertexBuffer (GraphicsDevice, Type, Int32, BufferUsage)	Initializes a new instance of VertexBuffer , specifying the element count and usage.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer Constructor (GraphicsDevice, Int32, BufferUsage)

Initializes a new instance of **VertexBuffer**, specifying the size and usage.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexBuffer (  
    GraphicsDevice graphicsDevice,  
    int sizeInBytes,  
    BufferUsage usage  
)
```

Parameters

graphicsDevice

The graphics device to associate with this vertex buffer.

sizeInBytes

The number of bytes to allocate for this vertex buffer.

usage

Options identifying the behaviors of this vertex buffer resource.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>graphicsDevice</i> cannot be null when creating new resources.
ArgumentOutOfRangeException	<i>sizeInBytes</i> must be greater than zero.
InvalidOperationException	This resource could not be created.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer Constructor (GraphicsDevice, Type, Int32, BufferUsage)

Initializes a new instance of **VertexBuffer**, specifying the element count and usage.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexBuffer (  
    GraphicsDevice graphicsDevice,  
    Type vertexType,  
    int elementCount,  
    BufferUsage usage  
)
```

Parameters

graphicsDevice

The graphics device to associate with this vertex buffer.

vertexType

The type of vertices in this vertex buffer.

elementCount

The number of elements in this vertex buffer.

usage

Options identifying the behaviors of this vertex buffer resource.

Exceptions

Exception type	Condition
ArgumentNullException	The <i>graphicsDevice</i> cannot be null when creating new resources.
ArgumentOutOfRangeException	<i>elementCount</i> must be greater than zero.
InvalidOperationException	This resource could not be created.

See Also

Reference

[VertexBuffer Class](#)









[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetData	Overloaded. Returns a copy of the vertex buffer data.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetData	Overloaded. Sets the vertex buffer data.
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	(Inherited from GraphicsResource .)

See Also

Reference

[VertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
VertexBuffer.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.
VertexBuffer.Dispose ()	(Inherited from GraphicsResource .)

Remarks

Call [Dispose](#) when you are finished using the [VertexBuffer](#). The [Dispose](#) method leaves the [VertexBuffer](#) in an unusable state. After calling [Dispose](#), you must release all references to the [VertexBuffer](#) so the garbage collector can reclaim the memory that the [VertexBuffer](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [VertexBuffer](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [VertexBuffer](#) object's **Finalize** method.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the **Finalize** method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. **Finalize** invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [VertexBuffer](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.GetData Method

Returns a copy of the vertex buffer data.

Overload List

Name	Description
VertexBuffer.GetData (Int32, T[], Int32, Int32, Int32)	Gets a copy of the vertex buffer data, specifying the start index, starting offset, number of elements, and size of the vertex buffer elements.
VertexBuffer.GetData (T[])	Gets a copy of the vertex buffer data.
VertexBuffer.GetData (T[], Int32, Int32)	Gets a copy of the vertex buffer data, specifying the start index and number of elements in the vertex buffer.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer.GetData Generic Method (Int32, T[], Int32, Int32, Int32)

Gets a copy of the vertex buffer data, specifying the start index, starting offset, number of elements, and size of the vertex buffer elements.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    int offsetInBytes,
    T[] data,
    int startIndex,
    int elementCount,
    int vertexStride
) where T : ValueType
```

Type Parameters

T

The type of data in the vertex buffer.

Parameters

offsetInBytes

Starting offset.

data

The array into which to copy the vertex buffer data.

startIndex

Index of the element in the array at which to start copying.

elementCount

Number of elements in the array.

vertexStride

Size, in bytes, of an element in the vertex buffer.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null . <i>data</i> must be of sufficient length to receive the data.
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The vertex stride is larger than the vertex buffer. The vertex stride is too small for the type of data requested.
InvalidOperationException	The array is not the correct size for the amount of data requested.
NotSupportedException	This VertexBuffer was created with a usage type of BufferUsage.WriteOnly . Calling GetData on a resource that was created with BufferUsage.WriteOnly is not supported.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.GetData Generic Method (T[])

Gets a copy of the vertex buffer data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    T[] data
) where T : ValueType
```

Type Parameters

T

The type of data in the vertex buffer.

Parameters

data

The array into which to copy the vertex buffer data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null . <i>data</i> must be of sufficient length to receive the data.
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The vertex stride is larger than the vertex buffer. The vertex stride is too small for the type of data requested.
InvalidOperationException	The array is not the correct size for the amount of data requested.
NotSupportedException	This VertexBuffer was created with a usage type of BufferUsage.WriteOnly . Calling GetData on a resource that was created with BufferUsage.WriteOnly is not supported.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.GetData Generic Method (T[], Int32, Int32)

Gets a copy of the vertex buffer data, specifying the start index and number of elements in the vertex buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void GetData<T> (
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of data in the vertex buffer.

Parameters

data

The array into which to copy the vertex buffer data.

startIndex

Index of the element in the array at which to start copying.

elementCount

Number of elements in the array.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null . <i>data</i> must be of sufficient length to receive the data.
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The vertex stride is larger than the vertex buffer. The vertex stride is too small for the type of data requested.
InvalidOperationException	The array is not the correct size for the amount of data requested.
NotSupportedException	This VertexBuffer was created with a usage type of BufferUsage.WriteOnly . Calling GetData on a resource that was created with BufferUsage.WriteOnly is not supported.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.SetData Method

Sets the vertex buffer data.

Overload List

Name	Description
VertexBuffer.SetData (Int32, T[], Int32, Int32, Int32)	Sets the vertex buffer data.
VertexBuffer.SetData (T[])	Sets the vertex buffer data.
VertexBuffer.SetData (T[], Int32, Int32)	Sets the vertex buffer data.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer.SetData Generic Method (Int32, T[], Int32, Int32, Int32)

Sets the vertex buffer data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    int offsetInBytes,
    T[] data,
    int startIndex,
    int elementCount,
    int vertexStride
) where T : ValueType
```

Type Parameters

T

The type of data in the vertex buffer.

Parameters

offsetInBytes

Starting offset.

data

Array from which to copy the vertex buffer data.

startIndex

Index of the element in the array at which to start copying.

elementCount

Number of elements to copy.

vertexStride

Size, in bytes, of an element in the vertex buffer.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The vertex stride is larger than the vertex buffer. The vertex stride is too small for the type of data requested.
InvalidOperationException	The array is not the correct size for the amount of data requested.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.SetData Generic Method (T[])

Sets the vertex buffer data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    T[] data
) where T : ValueType
```

Type Parameters

T

The type of data in the vertex buffer.

Parameters

data

The array from which to copy the vertex buffer data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The vertex stride is larger than the vertex buffer. The vertex stride is too small for the type of data requested.
InvalidOperationException	The array is not the correct size for the amount of data requested.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

⚠ Caution

Here is a tip for rendering objects within the [Draw](#) method of an Xbox 360 game. Do not use **SetData** when writing data to vertex buffers, index buffers, and textures. This method may lead to graphics corruption or crashes. To avoid this potential issue, use [DrawUserPrimitives](#), [DrawUserIndexedPrimitives](#), or [DynamicVertexBuffer](#) as the preferred alternative to [VertexBuffer.SetData](#) for dynamic vertex generation.

This is because, in cases where the size of the back buffer and depth stencil buffer exceed the size of the Xbox 360 10 MB of embedded memory (EDRAM), [predicated tiling](#) is utilized on this platform to compensate for the additional memory requirements. Predicated tiling is a process by which scene rendering is performed multiple times on subsections of the final render target dimensions.

When predicated tiling has been triggered, the drawing commands contained in the [Draw](#) function are not submitted until [Present](#) is called. (Note that [Draw](#) implicitly calls [Present](#) at the end of this method.) In this case, these resources are not available for modification until the GPU is finished with presenting the entire frame.

Example

This example demonstrates how to create a vertex buffer to hold a list of vertices and set the data for the vertex buffer to the array of vertices.

C#

```
vertexBuffer = new VertexBuffer(graphics.GraphicsDevice,
    VertexPositionNormalTexture.SizeInBytes * (pointList.Length),
    BufferUsage.None);

// Set the vertex buffer data to the array of vertices.
vertexBuffer.SetData<VertexPositionNormalTexture>(pointList);
```

Prior to drawing the vertices, set the source of the vertex stream for the graphics device by setting the desired vertex stream

from [GraphicsDevice.Vertices](#). Here, we use vertex stream zero, so **GraphicsDevice.Vertices[0]** is set to the vertex buffer created in step (1).

C#

```
graphics.GraphicsDevice.Vertices[0].SetSource(  
    vertexBuffer, 0,  
    VertexPositionNormalTexture.SizeInBytes);
```

See Also

Concepts

[Predicated Tiling](#)

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.SetData Generic Method (T[], Int32, Int32)

Sets the vertex buffer data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetData<T> (
    T[] data,
    int startIndex,
    int elementCount
) where T : ValueType
```

Type Parameters

T

The type of data in the vertex buffer.

Parameters

data

Array from which to copy the vertex buffer data.

startIndex

Index of the element in the array at which to start copying.

elementCount

Number of elements to copy.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
ArgumentOutOfRangeException	One of the following conditions is true: <ul style="list-style-type: none"> The vertex stride is larger than the vertex buffer. The vertex stride is too small for the type of data requested.
InvalidOperationException	The array is not the correct size for the amount of data requested.

Remarks An [InvalidOperationException](#) is thrown if an attempt is made to modify (for example, calls to the **SetData** method) a resource that is currently set on a graphics device.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexBuffer Class](#)









[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer Properties

Public Properties

	Name	Description
	BufferUsage	Gets the state of the related BufferUsage enumeration.
	GraphicsDevice	(Inherited from GraphicsResource .)
	IsDisposed	(Inherited from GraphicsResource .)
	Name	(Inherited from GraphicsResource .)
	Priority	(Inherited from GraphicsResource .)
	ResourceType	(Inherited from GraphicsResource .)
	SizeInBytes	Gets the size, in bytes, of this vertex buffer.
	Tag	(Inherited from GraphicsResource .)

See Also

Reference

[VertexBuffer Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexBuffer.BufferUsage Property

Gets the state of the related [BufferUsage](#) enumeration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public BufferUsage BufferUsage { get; }
```

Property Value

Indicates how the application uses buffer contents.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexBuffer.SizeInBytes Property

Gets the size, in bytes, of this vertex buffer.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int SizeInBytes { get; }
```

Property Value

The size, in bytes, of the vertex buffer.

See Also

Reference

[VertexBuffer Class](#)

[VertexBuffer Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration Class

Represents a vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class VertexDeclaration : IDisposable
```

See Also

Reference

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





VertexDeclaration Members

The following tables list the members exposed by the VertexDeclaration type.









Public Constructors

Name	Description
 VertexDeclaration	Initializes a new instance of the VertexDeclaration class.




Public Properties

Name	Description
 GraphicsDevice	Gets the GraphicsDevice associated with this vertex declaration.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Name	Returns the name of this vertex declaration.
 Tag	Returns the resource tags for this vertex declaration.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetVertexElements	Gets the vertex shader declaration.
 GetVertexStrideSize	Overloaded. Gets the size of a vertex from the vertex declaration.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[VertexDeclaration Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexDeclaration Constructor

Initializes a new instance of the [VertexDeclaration](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexDeclaration (  
    GraphicsDevice graphicsDevice,  
    VertexElement[] elements  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) associated with the vertex elements.

elements

An array of vertex elements.

Exceptions

Exception type	Condition
ArgumentNullException	<i>graphicsDevice</i> or <i>elements</i> is null .
ArgumentOutOfRangeException	<i>elements</i> does not contain any elements, or is too large for the graphics device.
InvalidOperationException	Unable to create this VertexDeclaration resource on the graphics device.

Remarks

VertexDeclaration objects must be recreated when the device is reset.

See Also

Tasks

[How To: Load Content](#)

Reference

[GraphicsDeviceManager.DeviceReset](#) Event

[VertexDeclaration](#) Class









[VertexDeclaration](#) Members

[Microsoft.Xna.Framework.Graphics](#) Namespace




Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetVertexElements	Gets the vertex shader declaration.
	GetVertexStrideSize	Overloaded. Gets the size of a vertex from the vertex declaration.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[VertexDeclaration Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexDeclaration.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
VertexDeclaration.Dispose ()	Immediately releases the unmanaged resources used by this object.
VertexDeclaration.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexDeclaration.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [VertexDeclaration](#). The [Dispose](#) method leaves the [VertexDeclaration](#) in an unusable state. After calling [Dispose](#), you must release all references to the [VertexDeclaration](#) so the garbage collector can reclaim the memory that the [VertexDeclaration](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [VertexDeclaration](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [VertexDeclaration](#) object's [Finalize](#) method.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [VertexDeclaration](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.GetVertexElements Method

Gets the vertex shader declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexElement[] GetVertexElements ()
```

Return Value

The array of vertex elements that make up the vertex shader declaration.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.GetVertexStrideSize Method

Gets the size of a vertex from the vertex declaration.

Overload List

Name	Description
VertexDeclaration.GetVertexStrideSize (Int32)	Gets the size of a vertex from the vertex declaration.
VertexDeclaration.GetVertexStrideSize (VertexElement[], Int32)	Gets the size of a vertex from the vertex declaration.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexDeclaration.GetVertexStrideSize Method (Int32)

Gets the size of a vertex from the vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int GetVertexStrideSize (  
    int stream  
)
```

Parameters

stream

The zero-based stream index.

Return Value

The vertex declaration size, in bytes.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.GetVertexStrideSize Method (VertexElement[], Int32)

Gets the size of a vertex from the vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static int GetVertexStrideSize (  
    VertexElement[] elements,  
    int stream  
)
```

Parameters

elements

The vertex declaration.

stream

The zero-based stream index.

Return Value

The vertex declaration size, in bytes.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the Disposing event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexDeclaration.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexDeclaration Class](#)





[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the GraphicsDevice associated with this vertex declaration.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Name	Returns the name of this vertex declaration.
	Tag	Returns the resource tags for this vertex declaration.

See Also

Reference

[VertexDeclaration Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexDeclaration.GraphicsDevice Property

Gets the [GraphicsDevice](#) associated with this vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The [GraphicsDevice](#) associated with this vertex declaration.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.Name Property

Returns the name of this vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of this vertex declaration.

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration.Tag Property

Returns the resource tags for this vertex declaration.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tags for this vertex declaration.

See Also

Reference

[VertexDeclaration Class](#)


[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexDeclaration Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[VertexDeclaration Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexDeclaration.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[VertexDeclaration Class](#)

[VertexDeclaration Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement Structure

Defines input vertex data to the pipeline.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct VertexElement
```

See Also

Reference

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista







VertexElement Members

The following tables list the members exposed by the VertexElement type.










Public Constructors

Name	Description
 VertexElement	Initializes a new instance of the VertexElement class.



Public Properties

Name	Description
 Offset	Retrieves or sets the offset (if any) from the beginning of the stream to the beginning of the vertex data.
 Stream	Retrieves or sets the stream number (or index) to use.
 UsageIndex	Modifies the usage data to allow the user to specify multiple usage types.
 VertexElementFormat	Gets or sets the format of this vertex element.
 VertexElementMethod	Gets or sets a value indicating which vertex data to calculate during tessellation.
 VertexElementUsage	Gets or sets a value describing how the vertex element is to be used.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
  Op_Equality	Compares two objects to determine whether they are the same.
  Op_Inequality	Compares two objects to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexElement Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexElement Constructor

Initializes a new instance of the [VertexElement](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexElement (
    short stream,
    short offset,
    VertexElementFormat elementFormat,
    VertexElementMethod elementMethod,
    VertexElementUsage elementUsage,
    byte usageIndex
)
```

Parameters

stream

Stream number (or index) to use.

offset

Offset (if any) from the beginning of the stream to the beginning of the vertex data.

elementFormat

One of several predefined types that define the vertex data size.

elementMethod

The tessellator processing method. These methods determine how the tessellator interprets/operates on the vertex data.

elementUsage

The intended use of the vertex data.

usageIndex

Modifies the usage data to allow the user to specify multiple usage types.

Remarks Vertex data is defined using an array of [VertexElement](#) structures.

See Also

Reference

[VertexElement Structure](#)








[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares two objects to determine whether they are the same.
	Op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexElement Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexElement.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
VertexElement.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
VertexElement.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexElement.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [VertexElement](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexElement left,  
    VertexElement right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexElement left,  
    VertexElement right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexElement Structure](#)







[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement Properties

Public Properties

	Name	Description
	Offset	Retrieves or sets the offset (if any) from the beginning of the stream to the beginning of the vertex data.
	Stream	Retrieves or sets the stream number (or index) to use.
	UsageIndex	Modifies the usage data to allow the user to specify multiple usage types.
	VertexElementFormat	Gets or sets the format of this vertex element.
	VertexElementMethod	Gets or sets a value indicating which vertex data to calculate during tessellation.
	VertexElementUsage	Gets or sets a value describing how the vertex element is to be used.

See Also

Reference

[VertexElement Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexElement.Offset Property

Retrieves or sets the offset (if any) from the beginning of the stream to the beginning of the vertex data.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public short Offset { get; set; }
```

Property Value

Offset, in bytes, to the start of the vertex data.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.Stream Property

Retrieves or sets the stream number (or index) to use.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public short Stream { get; set; }
```

Property Value

Stream number (or index) to use.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.UsageIndex Property

Modifies the usage data to allow the user to specify multiple usage types.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte UsageIndex { get; set; }
```

Property Value

Byte that represents the usage index.

RemarksThe **UsageIndex** property is used in conjunction with [VertexElementUsage](#) to specify the semantic of a vertex element.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista

VertexElement.VertexElementFormat Property

Gets or sets the format of this vertex element.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexElementFormat VertexElementFormat { get; set; }
```

Property Value

The format of this vertex element.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.VertexElementMethod Property

Gets or sets a value indicating which vertex data to calculate during tessellation.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexElementMethod VertexElementMethod { get; set; }
```

Property Value

A value indicating which vertex data to calculate during tessellation.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElement.VertexElementUsage Property

Gets or sets a value describing how the vertex element is to be used.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexElementUsage VertexElementUsage { get; set; }
```

Property Value

A value describing how the vertex element is to be used.

See Also

Reference

[VertexElement Structure](#)

[VertexElement Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElementFormat Enumeration

Defines vertex element formats.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum VertexElementFormat
```

Members

Member name	Description
Single	Single-component, 32-bit floating-point, expanded to <i>(float, 0, 0, 1)</i> .
Vector2	Two-component, 32-bit floating-point, expanded to <i>(float, Float32 value, 0, 1)</i> .
Vector3	Three-component, 32-bit floating point, expanded to <i>(float, float, float, 1)</i> .
Vector4	Four-component, 32-bit floating point, expanded to <i>(float, float, float, float)</i> .
HalfVector2	Two-component, 16-bit floating point expanded to <i>(value, value, value, value)</i> . This type is valid for vertex shader version 2.0 or higher.
HalfVector4	Four-component, 16-bit floating-point expanded to <i>(value, value, value, value)</i> . This type is valid for vertex shade r version 2.0 or higher.
Rgba64	Normalized, four-component, unsigned short, expanded to <i>(first byte/65535.0, second byte/65535.0, third byte/65535.0, fourth byte/65535.0)</i> . This type is valid for vertex shader version 2.0 or higher.
Color	Four-component, packed, unsigned byte, mapped to 0 to 1 range. Input is in Int32 format (ARGB) expanded to (R, G, B, A).
Rgba32	Four-component byte with each byte normalized by dividing the component with 255.0f. This type is valid for vertex shader version 2.0 or higher.
Rg32	Normalized, two-component, unsigned short, expanded to <i>(first byte/65535.0, second byte/65535.0, 0, 1)</i> . This type is valid for vertex shader version 2.0 or higher.
NormalizedShort2	Normalized, two-component, signed short, expanded to <i>(first short/32767.0, second short/32767.0, 0, 1)</i> . This type is valid for vertex shader version 2.0 or higher.
NormalizedShort4	Normalized, four-component, signed short, expanded to <i>(first short/32767.0, second short/32767.0, third short/32767.0, fourth short/32767.0)</i> . This type is valid for vertex shader version 2.0 or higher.
Normalized101010	Three-component, signed, 10 10 10 format normalized and expanded to <i>(v[0]/511.0, v[1]/511.0, v[2]/511.0, 1)</i> .
Short2	Two-component, signed short expanded to <i>(value, value, 0, 1)</i> .
Short4	Four-component, signed short expanded to <i>(value, value, value, value)</i> .
Byte4	Four-component, unsigned byte.
UInt101010	Three-component, unsigned, 10 10 10 format expanded to <i>(value, value, value, 1)</i> .
Unused	Type field in the declaration is unused. This is designed for use with VertexElementMethod.UV and VertexElementMethod.LookUpPresampled .

Remarks

Vertex data is declared with an array of [VertexElement](#) structures. Each element in the array contains a vertex declaration method. These values can be passed into the constructor or to the [VertexElementMethod](#) property.

A mapping of each Direct3D 9 format name to the **VertexElementFormat** equivalent is listed in the following table.

	Direct3D 9 Vertex Format	VertexElementFormat equivalent
Floating Point		
Float32	D3DDECLTYPE_FLOAT1	Single
	D3DDECLTYPE_FLOAT2	Vector2
	D3DDECLTYPE_FLOAT3	Vector3
	D3DDECLTYPE_FLOAT4	Vector4
Float16	D3DDECLTYPE_FLOAT16_2	HalfVector2
	D3DDECLTYPE_FLOAT16_4	HalfVector4

Unsigned Normalized		
64 bpp	D3DDECLTYPE_USHORT4N	Rgba64
32 bpp	D3DDECLTYPE_D3DCOLOR	Color
	D3DDECLTYPE_UBYTE4N	Rgba32
	D3DDECLTYPE_USHORT2N	Rg32
Signed Normalized		
	D3DDECLTYPE_SHORT2N	NormalizedShort2
	D3DDECLTYPE_SHORT4N	NormalizedShort4
	D3DDECLTYPE_DEC3N	Normalized101010
Signed Integer		
	D3DDECLTYPE_SHORT2	Short2
	D3DDECLTYPE_SHORT4	Short4
Unsigned Integer		
	D3DDECLTYPE_UBYTE4	Byte4
	D3DDECLTYPE_UDEC3	UInt101010

See Also

Reference

[VertexElement Constructor](#)

[VertexElement.VertexElementFormat Property](#)

[GraphicsDeviceCapabilities.DeclarationTypeCaps Properties](#)

[VertexElementMethod Enumeration](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElementMethod Enumeration

Defines the tessellator processing method for a vertex element.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum VertexElementMethod
```

Members

Member name	Description
Default	Default value. The tessellator copies the vertex data (or the spline data if it is operating on a patch) without performing additional calculations on it. The input and output types can be any value. When the tessellator is used, this element is interpolated; otherwise, vertex data is copied into the input register.
Lookup	Looks up a displacement map. The input type can be VertexElementFormat.Vector2 , VertexElementFormat.Vector3 , or VertexElementFormat.Vector4 . Only the .x and .y components are used for the texture map lookup. The output type is always VertexElementFormat.Single . The device must support displacement mapping. This constant is supported only by the programmable pipeline on N-patch data, if N-patches are enabled.
LookupPresampled	Looks up a presampled displacement map. The input type must be set to VertexElementFormat.Unused , and the stream index and stream offset must be set to 0. The output type for this operation is always VertexElementFormat.Single . The device must support displacement mapping. This constant is supported only by the programmable pipeline on N-patch data, if N-patches are enabled.
UV	Copies out the u and v values at a point on the rectangle or triangle patch. This results in a 2D float. The input type must be set to VertexElementFormat.Unused ; the output type is always VertexElementFormat.Vector2 . The input stream and offset also are unused, but must be set to 0.

Remarks

The [VertexElement](#) structure uses **VertexElementMethod** values to determine which vertex data to calculate during tessellation. These values can be passed into the constructor or to the [VertexElementMethod](#) property. Mesh data should use the default value. Patches can use any of the other implemented types.

Vertex data is declared with an array of [VertexElement](#) structures. Each element in the array contains a vertex declaration method.

In addition to using **Default**, a normal mesh can use **Lookup** and **LookupPresampled** when N-patches are enabled.

See Also

Reference

[VertexElement Constructor](#)

[VertexElement.VertexElementMethod Property](#)

[VertexElementFormat Enumeration](#)

[VertexElementUsage Enumeration](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexElementUsage Enumeration

Defines usage for vertex elements.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum VertexElementUsage
```

Members

Member name	Description
Binormal	Vertex binormal data.
BlendIndices	Blending indices data. (BlendIndices with UsageIndex = 0) specifies matrix indices for fixed-function vertex processing using indexed paletted skinning.
BlendWeight	Blending weight data. (BlendWeight with UsageIndex = 0) specifies the blend weights in fixed-function vertex processing.
Color	Vertex data contains diffuse or specular color. (Color with UsageIndex = 0) specifies the diffuse color in the fixed-function vertex shader and in pixel shaders prior to ps_3_0. (Color with UsageIndex = 1) specifies the specular color in the fixed-function vertex shader and in pixel shaders prior to ps_3_0.
Depth	Vertex data contains depth data.
Fog	Vertex data contains fog data. (Fog with UsageIndex = 0) specifies a fog blend value to use after pixel shading is finished. This flag applies to pixel shaders prior to version ps_3_0.
Normal	Vertex normal data. (Normal with UsageIndex = 0) specifies vertex normals for fixed-function vertex processing and the N-patch tessellator. (Normal with UsageIndex = 1) specifies vertex normals for fixed-function vertex processing for skinning.
PointSize	Point size data. (PointSize with UsageIndex = 0) specifies the point-size attribute used by the setup engine of the rasterizer to expand a point into a quad for the point-sprite functionality.
Position	Position data. (Position with UsageIndex = 0) specifies the nontransformed position in fixed-function vertex processing and the N-patch tessellator. (Position with UsageIndex = 1) specifies the nontransformed position in the fixed-function vertex shader for skinning.
Sampler	Vertex data contains sampler data. (Sampler with UsageIndex = 0) specifies the displacement value to look up. This flag can be used only with VertexElementMethod.LookUpPresampled or VertexElementMethod.LookUp .
Tangent	Vertex tangent data.
TessellateFactor	Single, positive floating-point value. (TessellateFactor with UsageIndex = 0) specifies a tessellation factor used in the tessellation unit to control the rate of tessellation.
TextureCoordinate	Texture coordinate data. (TextureCoordinate , <i>n</i>) specifies texture coordinates in fixed-function vertex processing and in pixel shaders prior to ps_3_0. These coordinates can be used to pass user-defined data.

Remarks Vertex data is declared with an array of [VertexElement](#) structures. Each element in the array contains a usage type.

See Also

Reference

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor Structure

Describes a custom vertex format structure that contains position and color information.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct VertexPositionColor
```

See Also

Reference

[VertexPositionColor Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





VertexPositionColor Members

The following tables list the members exposed by the VertexPositionColor type.



Public Constructors

	Name	Description
	VertexPositionColor	Initializes a new instance of the VertexPositionColor class.










Public Fields

	Name	Description
	Color	The vertex color.
	Position	The vertex position.
 	VertexElements	An array of two vertex elements describing the position, followed by the color, of this vertex.



Public Properties

	Name	Description
 	SizeInBytes	Gets the size of the VertexPositionColor class.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
 	op_Equality	Compares two objects to determine whether they are the same.
 	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[VertexPositionColor Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColor Fields

Public Fields

	Name	Description
	Color	The vertex color.
	Position	The vertex position.
 	VertexElements	An array of two vertex elements describing the position, followed by the color, of this vertex.

See Also

Reference

[VertexPositionColor Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColor.Color Field

The vertex color.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color Color
```

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor.Position Field

The vertex position.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position
```

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor.VertexElements Field

An array of two vertex elements describing the position, followed by the color, of this vertex.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static readonly VertexElement[] VertexElements
```

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor Constructor

Initializes a new instance of the [VertexPositionColor](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexPositionColor (  
    Vector3 position,  
    Color color  
)
```

Parameters

position

The position of the vertex.

color

The color of the vertex.

See Also

Reference

[VertexPositionColor Structure](#)








[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexPositionColor Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColor.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
VertexPositionColor.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
VertexPositionColor.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColor.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [VertexPositionColor](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexPositionColor left,  
    VertexPositionColor right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexPositionColor left,  
    VertexPositionColor right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexPositionColor Structure](#)


[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColor Properties

Public Properties

	Name	Description
	S SizeInBytes	Gets the size of the VertexPositionColor class.

See Also

Reference

[VertexPositionColor Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColor.SizeInBytes Property

Gets the size of the [VertexPositionColor](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static int SizeInBytes { get; }
```

Property Value

The size of the vertex, in bytes.

See Also

Reference

[VertexPositionColor Structure](#)

[VertexPositionColor Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture Structure

Describes a custom vertex format structure that contains position, color, and one set of texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct VertexPositionColorTexture
```

See Also

Reference

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






VertexPositionColorTexture Members

The following tables list the members exposed by the VertexPositionColorTexture type.



Public Constructors

	Name	Description
	VertexPositionColorTexture	Initializes a new instance of the VertexPositionColorTexture class.










Public Fields

	Name	Description
	Color	The vertex color.
	Position	The vertex position.
	TextureCoordinate	The texture coordinates.
	 VertexElements	An array of three vertex elements describing the position, texture coordinate, and color of this vertex.



Public Properties

	Name	Description
	 SizeInBytes	Gets the size of the VertexPositionColorTexture class.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares two objects to determine whether they are the same.
	 op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[VertexPositionColorTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColorTexture Fields

Public Fields

	Name	Description
	Color	The vertex color.
	Position	The vertex position.
	TextureCoordinate	The texture coordinates.
 S	VertexElements	An array of three vertex elements describing the position, texture coordinate, and color of this vertex.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColorTexture.Color Field

The vertex color.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color Color
```

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.Position Field

The vertex position.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position
```

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.TextureCoordinate Field

The texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 TextureCoordinate
```

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.VertexElements Field

An array of three vertex elements describing the position, texture coordinate, and color of this vertex.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static readonly VertexElement[] VertexElements
```

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture Constructor

Initializes a new instance of the [VertexPositionColorTexture](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexPositionColorTexture (  
    Vector3 position,  
    Color color,  
    Vector2 textureCoordinate  
)
```

Parameters

position

Position of the vertex.

color

Color of the vertex.

textureCoordinate

Texture coordinate of the vertex.

See Also

Reference

[VertexPositionColorTexture Structure](#)








[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares two objects to determine whether they are the same.
	Op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexPositionColorTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColorTexture.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
VertexPositionColorTexture.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
VertexPositionColorTexture.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColorTexture.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [VertexPositionColorTexture](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexPositionColorTexture left,  
    VertexPositionColorTexture right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexPositionColorTexture left,  
    VertexPositionColorTexture right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexPositionColorTexture Structure](#)


[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionColorTexture Properties

Public Properties

	Name	Description
 S	SizeInBytes	Gets the size of the VertexPositionColorTexture class.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionColorTexture.SizeInBytes Property

Gets the size of the [VertexPositionColorTexture](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static int SizeInBytes { get; }
```

Property Value

The size of the vertex, in bytes.

See Also

Reference

[VertexPositionColorTexture Structure](#)

[VertexPositionColorTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture Structure

Describes a custom vertex format structure that contains position, normal data, and one set of texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct VertexPositionNormalTexture
```

See Also

Tasks

[How To: Use BasicEffect](#)

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[VertexPositionNormalTexture Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






VertexPositionNormalTexture Members

The following tables list the members exposed by the VertexPositionNormalTexture type.



Public Constructors

	Name	Description
	VertexPositionNormalTexture	Initializes a new instance of the VertexPositionNormalTexture class.










Public Fields

	Name	Description
	Normal	The vertex normal.
	Position	The vertex position.
	TextureCoordinate	The texture coordinates.
 	VertexElements	An array of three vertex elements describing the position, normal, and texture coordinate of this vertex.



Public Properties

	Name	Description
 	SizeInBytes	Gets the size of the VertexPositionNormalTexture class.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
 	Op_Equality	Compares two objects to determine whether they are the same.
 	Op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[VertexPositionNormalTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionNormalTexture Fields

Public Fields

	Name	Description
	Normal	The vertex normal.
	Position	The vertex position.
	TextureCoordinate	The texture coordinates.
 	VertexElements	An array of three vertex elements describing the position, normal, and texture coordinate of this vertex.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionNormalTexture.Normal Field

The vertex normal.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Normal
```

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.Position Field

The vertex position.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position
```

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.TextureCoordinate Field

The texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 TextureCoordinate
```

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.VertexElements Field

An array of three vertex elements describing the position, normal, and texture coordinate of this vertex.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static readonly VertexElement[] VertexElements
```

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture Constructor

Initializes a new instance of the [VertexPositionNormalTexture](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexPositionNormalTexture (  
    Vector3 position,  
    Vector3 normal,  
    Vector2 textureCoordinate  
)
```

Parameters

position

Position of the vertex.

normal

The vertex normal.

textureCoordinate

The texture coordinate.

See Also

Reference

[VertexPositionNormalTexture Structure](#)








[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionNormalTexture.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
VertexPositionNormalTexture.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
VertexPositionNormalTexture.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionNormalTexture.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [VertexPositionNormalTexture](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexPositionNormalTexture left,  
    VertexPositionNormalTexture right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexPositionNormalTexture left,  
    VertexPositionNormalTexture right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexPositionNormalTexture Structure](#)


[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionNormalTexture Properties

Public Properties

	Name	Description
	S SizeInBytes	Gets the size of the VertexPositionNormalTexture class.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionNormalTexture.SizeInBytes Property

Gets the size of the [VertexPositionNormalTexture](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static int SizeInBytes { get; }
```

Property Value

The size of the vertex, in bytes.

See Also

Reference

[VertexPositionNormalTexture Structure](#)

[VertexPositionNormalTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture Structure

Describes a custom vertex format structure that contains position and one set of texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct VertexPositionTexture
```

See Also

Tasks

[How To: Create Custom Texture Effects](#)

Reference

[VertexPositionTexture Members](#)


[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





VertexPositionTexture Members

The following tables list the members exposed by the VertexPositionTexture type.



Public Constructors

	Name	Description
	VertexPositionTexture	Initializes a new instance of the VertexPositionTexture class.










Public Fields

	Name	Description
	Position	The vertex position.
	TextureCoordinate	The texture coordinates.
 	VertexElements	An array of two vertex elements describing the position, followed by the texture coordinate, of this vertex.



Public Properties

	Name	Description
 	SizeInBytes	Gets the size of the VertexPositionTexture class.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
 	op_Equality	Compares two objects to determine whether they are the same.
 	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[VertexPositionTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionTexture Fields

Public Fields

	Name	Description
	Position	The vertex position.
	TextureCoordinate	The texture coordinates.
 	VertexElements	An array of two vertex elements describing the position, followed by the texture coordinate, of this vertex.

See Also

Reference

[VertexPositionTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionTexture.Position Field

The vertex position.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Position
```

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture.TextureCoordinate Field

The texture coordinates.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 TextureCoordinate
```

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture.VertexElements Field

An array of two vertex elements describing the position, followed by the texture coordinate, of this vertex.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static readonly VertexElement[] VertexElements
```

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture Constructor

Initializes a new instance of the [VertexPositionTexture](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexPositionTexture (  
    Vector3 position,  
    Vector2 textureCoordinate  
)
```

Parameters

position

Position of the vertex.

textureCoordinate

Texture coordinate of the vertex.

See Also

Reference

[VertexPositionTexture Structure](#)








[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares two objects to determine whether they are the same.
	op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexPositionTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionTexture.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
VertexPositionTexture.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
VertexPositionTexture.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionTexture.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The [Object](#) to compare with the current [VertexPositionTexture](#).

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    VertexPositionTexture left,  
    VertexPositionTexture right  
)
```

Parameters

left

Object to the left of the equality operator.

right

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    VertexPositionTexture left,  
    VertexPositionTexture right  
)
```

Parameters

left

Object to the left of the inequality operator.

right

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexPositionTexture Structure](#)


[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexPositionTexture Properties

Public Properties

	Name	Description
	S SizeInBytes	Gets the size of the VertexPositionTexture class.

See Also

Reference

[VertexPositionTexture Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexPositionTexture.SizeInBytes Property

Gets the size of the [VertexPositionTexture](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static int SizeInBytes { get; }
```

Property Value

The size of the vertex, in bytes.

See Also

Reference

[VertexPositionTexture Structure](#)

[VertexPositionTexture Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader Class

Encapsulates the functionality of a vertex shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class VertexShader : IDisposable
```

See Also

Reference

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista





VertexShader Members

The following tables list the members exposed by the VertexShader type.








Public Constructors

Name	Description
 VertexShader	Initializes a new instance of the VertexShader class.




Public Properties

Name	Description
 GraphicsDevice	Gets the GraphicsDevice associated with this VertexShader .
 IsDisposed	Gets a value that indicates whether the object is disposed.
 Name	Gets the name of this vertex shader.
 Tag	Gets the resource tags for this vertex shader.


Public Methods

Name	Description
 Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetShaderCode	Gets the shader byte code.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)
 raise_Disposing	Raises the Disposing event when called from within a derived class.

Public Events

Name	Description
 Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[VertexShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexShader Constructor

Initializes a new instance of the [VertexShader](#) class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexShader (  
    GraphicsDevice graphicsDevice,  
    byte[] shaderCode  
)
```

Parameters

graphicsDevice

The [GraphicsDevice](#) to associate with this [VertexShader](#).

shaderCode

The compiled byte code. An array of tokens that represents the vertex shader, including embedded debug and symbol table information.

Exceptions

Exception type	Condition
ArgumentException	<i>shaderCode</i> must have a length that is a multiple of four.
ArgumentNullException	<i>graphicsDevice</i> or <i>shaderCode</i> is null .
InvalidOperationException	Unable to create this VertexShader resource on the graphics device.

See Also

Reference

[VertexShader Class](#)








[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)




Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader Methods

Public Methods

	Name	Description
	Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetShaderCode	Gets the shader byte code.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)
	raise_Disposing	Raises the Disposing event when called from within a derived class.

See Also

Reference

[VertexShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexShader.Dispose Method

Immediately releases the unmanaged resources used by this object.

Overload List

Name	Description
VertexShader.Dispose ()	Immediately releases the unmanaged resources used by this object.
VertexShader.Dispose (Boolean)	Immediately releases the unmanaged resources used by this object.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexShader.Dispose Method ()

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call [Dispose](#) when you are finished using the [VertexShader](#). The [Dispose](#) method leaves the [VertexShader](#) in an unusable state. After calling [Dispose](#), you must release all references to the [VertexShader](#) so the garbage collector can reclaim the memory that the [VertexShader](#) was occupying.

Note

Always call [Dispose](#) before you release your last reference to the [VertexShader](#). Otherwise, the resources it is using will not be freed until the garbage collector calls the [VertexShader](#) object's [Finalize](#) method.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.Dispose Method (Boolean)

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Dispose (  
    bool  
)
```

Parameters

[\[MarshalAsAttribute\(U1\)\]](#) **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

Remarks

This method is called by the public [Dispose](#) method and the [Finalize](#) method. [Dispose](#) invokes the protected [Dispose\(Boolean\)](#) method with the *disposing* parameter set to **true**. [Finalize](#) invokes [Dispose\(Boolean\)](#) with *disposing* set to **false**.

When the *disposing* parameter is **true**, this method releases all resources held by any managed objects that this [VertexShader](#) references. This method invokes the [Dispose](#) method of each referenced object.

Note

Notes to Inheritors

[Dispose](#) can be called multiple times by other objects. When overriding [Dispose\(Boolean\)](#), be careful not to reference objects disposed of in an earlier call to [Dispose](#).

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method; an object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.GetShaderCode Method

Gets the shader byte code.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte[] GetShaderCode ()
```

Return Value

The shader byte code.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.raise_Disposing Method

Note

This method is available only when developing for Windows.

Raises the Disposing event when called from within a derived class.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected void raise_Disposing (  
    Object value0,  
    EventArgs value1  
)
```

Parameters

value0

Invoking object reference; should be this object.

value1

Arguments to pass to the event handler.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexShader.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[VertexShader Class](#)





[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader Properties

Public Properties

	Name	Description
	GraphicsDevice	Gets the GraphicsDevice associated with this VertexShader .
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Name	Gets the name of this vertex shader.
	Tag	Gets the resource tags for this vertex shader.

See Also

Reference

[VertexShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexShader.GraphicsDevice Property

Gets the [GraphicsDevice](#) associated with this [VertexShader](#).

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GraphicsDevice GraphicsDevice { get; }
```

Property Value

The [GraphicsDevice](#) associated with this [VertexShader](#).

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.Name Property

Gets the name of this vertex shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of this vertex shader.

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader.Tag Property

Gets the resource tags for this vertex shader.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

The resource tags for this vertex shader.

See Also

Reference

[VertexShader Class](#)


[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexShader Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[VertexShader Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexShader.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[VertexShader Class](#)

[VertexShader Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexStream Class

Represents a vertex stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class VertexStream
```

See Also

Tasks

[How To: Draw Points, Lines, and Other 3D Primitives](#)

Reference

[VertexStream Members](#)




[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista










VertexStream Members

The following tables list the members exposed by the VertexStream type.



Public Properties

Name	Description
 OffsetInBytes	Gets the starting offset of the vertex stream.
 VertexBuffer	Gets the vertex buffer associated with this vertex stream.
 VertexStride	Gets the size, in bytes, of the elements in this vertex stream.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetFrequency	Sets the stream source frequency divider value. This may be used to draw several instances of geometry.
 SetFrequencyOfIndexData	Sets the stream source frequency divider value for the index data. This may be used to draw several instances of geometry.
 SetFrequencyOfInstanceData	Sets the stream source frequency divider value for the instance data. This may be used to draw several instances of geometry.
 SetSource	Sets the source of the vertex stream.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also










Reference

[VertexStream Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexStream Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetFrequency	Sets the stream source frequency divider value. This may be used to draw several instances of geometry.
	SetFrequencyOfIndexData	Sets the stream source frequency divider value for the index data. This may be used to draw several instances of geometry.
	SetFrequencyOfInstanceData	Sets the stream source frequency divider value for the instance data. This may be used to draw several instances of geometry.
	SetSource	Sets the source of the vertex stream.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexStream Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexStream.SetFrequency Method

Note

This method is available only when developing for Windows.

Sets the stream source frequency divider value. This may be used to draw several instances of geometry.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetFrequency (  
    int frequency  
)
```

Parameters

frequency

Frequency divider value.

See Also

Reference

[VertexStream Class](#)

[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexStream.SetFrequencyOfIndexData Method

Note

This method is available only when developing for Windows.

Sets the stream source frequency divider value for the index data. This may be used to draw several instances of geometry.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetFrequencyOfIndexData (  
    int frequency  
)
```

Parameters

frequency

Frequency of index data.

See Also

Reference

[VertexStream Class](#)

[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexStream.SetFrequencyOfInstanceData Method

Note

This method is available only when developing for Windows.

Sets the stream source frequency divider value for the instance data. This may be used to draw several instances of geometry.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetFrequencyOfInstanceData (  
    int frequency  
)
```

Parameters

frequency

Frequency of instance data.

See Also

Reference

[VertexStream Class](#)

[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexStream.SetSource Method

Sets the source of the vertex stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SetSource (  
    VertexBuffer vb,  
    int offsetInBytes,  
    int vertexStride  
)
```

Parameters

vb

The vertex buffer source.

offsetInBytes

The starting offset.

vertexStride

The size, in bytes, of the elements in the vertex buffer.

Exceptions

Exception type	Condition
InvalidOperationException	The requested vertex stream is invalid.
ObjectDisposedException	SetSource was called after the GraphicsDevice was disposed.

See Also

Reference

[VertexStream Class](#)




[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexStream Properties

Public Properties

	Name	Description
	OffsetInBytes	Gets the starting offset of the vertex stream.
	VertexBuffer	Gets the vertex buffer associated with this vertex stream.
	VertexStride	Gets the size, in bytes, of the elements in this vertex stream.

See Also

Reference

[VertexStream Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexStream.OffsetInBytes Property

Gets the starting offset of the vertex stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int OffsetInBytes { get; }
```

Property Value

Starting offset of the vertex stream.

See Also

Reference

[VertexStream Class](#)

[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexStream.VertexBuffer Property

Gets the vertex buffer associated with this vertex stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexBuffer VertexBuffer { get; }
```

Property Value

Vertex buffer associated with this vertex stream.

See Also

Reference

[VertexStream Class](#)

[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexStream.VertexStride Property

Gets the size, in bytes, of the elements in this vertex stream.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int VertexStride { get; }
```

Property Value

Size, in bytes, of the elements in this vertex stream.

See Also

Reference

[VertexStream Class](#)

[VertexStream Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VertexStreamCollection Class

Collection of [VertexStream](#) objects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class VertexStreamCollection
```

See Also

Reference

[VertexStreamCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






VertexStreamCollection Members

The following tables list the members exposed by the VertexStreamCollection type.



Public Properties

	Name	Description
	Item	Returns the VertexStream at the specified index.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[VertexStreamCollection Class](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexStreamCollection Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexStreamCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexStreamCollection Properties

Public Properties

	Name	Description
	Item	Returns the VertexStream at the specified index.

See Also

Reference

[VertexStreamCollection Class](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

VertexStreamCollection.Item Property

Returns the [VertexStream](#) at the specified index.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VertexStream this [
    int index
] { get; }
```

Property Value

The vertex stream at the requested index.

Exceptions

Exception type	Condition
InvalidOperationException	The specified <i>index</i> is invalid.

See Also

Reference

[VertexStreamCollection Class](#)

[VertexStreamCollection Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Viewport Structure

Defines the window dimensions of a render-target surface onto which a 3D volume projects.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public struct Viewport
```

See Also

Concepts

[Displays, Client Bounds, Viewports, and Back Buffers](#)

Reference

[Viewport](#)

[Viewport Members](#)









[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







Viewport Members

The following tables list the members exposed by the Viewport type.



Public Properties

Name	Description
 AspectRatio	Gets the aspect ratio used by the viewport
 Height	Gets or sets the height dimension of the viewport on the render-target surface, in pixels.
 MaxDepth	Gets or sets the maximum depth of the clip volume.
 MinDepth	Gets or sets the minimum depth of the clip volume.
 TitleSafeArea	Returns the title safe area of the current viewport.
 Width	Gets or sets the width dimension of the viewport on the render-target surface, in pixels.
 X	Gets or sets the pixel coordinate of the upper-left corner of the viewport on the render-target surface.
 Y	Gets or sets the pixel coordinate of the upper-left corner of the viewport on the render-target surface.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Project	Projects a 3D vector from object space into screen space.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.
 Unproject	Converts a screen space point into a corresponding point in world space.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also







Reference

[Viewport Structure](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

Viewport Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Project	Projects a 3D vector from object space into screen space.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.
	Unproject	Converts a screen space point into a corresponding point in world space.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Viewport Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Viewport.Project Method

Projects a 3D vector from object space into screen space.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Project (  
    Vector3 source,  
    Matrix projection,  
    Matrix view,  
    Matrix world  
)
```

Parameters

source

The vector to project.

projection

The projection matrix.

view

The view matrix.

world

The world matrix.

Return Value

The vector in screen space.

See Also

Tasks

[How To: Draw a Sprite Over a Model](#)

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.Unproject Method

Converts a screen space point into a corresponding point in world space.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Unproject (  
    Vector3 source,  
    Matrix projection,  
    Matrix view,  
    Matrix world  
)
```

Parameters

source

The vector to project.

projection

The projection matrix.

view

The view matrix.

world

The world matrix.

Return Value

The vector in object space.

Remarks

A common use for **Unproject** is determining if the current cursor location intersects with an object in 3D world space. First, calculate two [Vector3](#) values that differ only by their Z value. For instance, assume that the cursor location is currently (100, 100). Therefore, the first vector (located at the near clip plane) becomes (100, 100, 0) and the second (located at the far clip plane) becomes (100, 100, 1).

Call **Unproject** for each point, and store the result. For example, `minPointSource` stores the result of "unprojecting" (100,100,0), and `maxPointSource` stores the result of "unprojecting" (100, 100, 1). Determine the direction vector by subtracting `maxPointSource` from `minPointSource`.

Finally, normalize the direction vector, and create a [Ray](#) with `minPointSource` and the now-normalized direction vector. You can now use this ray in a simple intersect test case (for example, [Intersects](#)) with the model.

For more usage examples of **Unproject**, see the following:

- [How To: Detect Whether a User Clicked a 3D Object](#)
- Picking sample (found at <http://creators.xna.com>)
- Picking with Triangle Accuracy sample (also found at <http://creators.xna.com>)

See Also

Reference

[Viewport Structure](#)









[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport Properties

Public Properties

	Name	Description
	AspectRatio	Gets the aspect ratio used by the viewport
	Height	Gets or sets the height dimension of the viewport on the render-target surface, in pixels.
	MaxDepth	Gets or sets the maximum depth of the clip volume.
	MinDepth	Gets or sets the minimum depth of the clip volume.
	TitleSafeArea	Returns the title safe area of the current viewport.
	Width	Gets or sets the width dimension of the viewport on the render-target surface, in pixels.
	X	Gets or sets the pixel coordinate of the upper-left corner of the viewport on the render-target surface.
	Y	Gets or sets the pixel coordinate of the upper-left corner of the viewport on the render-target surface.

See Also

Reference

[Viewport Structure](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Viewport.AspectRatio Property

Gets the aspect ratio used by the viewport

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float AspectRatio { get; }
```

Property Value

The aspect ratio of the viewport.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.Height Property

Gets or sets the height dimension of the viewport on the render-target surface, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; set; }
```

Property Value

The height, in pixels, of the viewport to set or get.

RemarksUnless rendering is being done only to a subset of the surface, this member should be set to the height dimension of the render target surface.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.MaxDepth Property

Gets or sets the maximum depth of the clip volume.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MaxDepth { get; set; }
```

Property Value

The maximum depth of the clipping volume.

Remarks **MaxDepth**, together with [MinDepth](#), specifies the range of depth values into which a scene is rendered. Most applications set this value to 1.0. Clipping is performed after the projection matrix is applied.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.MinDepth Property

Gets or sets the minimum depth of the clip volume.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MinDepth { get; set; }
```

Property Value

The minimum depth of the clipping volume.

Remarks **MinDepth**, together with [MaxDepth](#), specifies the range of depth values into which a scene is rendered. Most applications set this value to 0.0. Clipping is performed after the projection matrix is applied.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.TitleSafeArea Property

Returns the title safe area of the current viewport.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rectangle TitleSafeArea { get; }
```

Property Value

The title safe area.

Remarks

On television sets, text should not be displayed at the outer edges of the screen because it may not be visible to the user. The title safe area is the part of the screen where text should be displayed.

This property does not compare the current location of the viewport with the title safe area of the entire display—it reduces the title safe area of the viewport uniformly.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.Width Property

Gets or sets the width dimension of the viewport on the render-target surface, in pixels.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; set; }
```

Property Value

The width, in pixels, of the viewport to set or get.

RemarksUnless rendering is being done only to a subset of the surface, this member should be set to the width dimension of the render-target surface.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.X Property

Gets or sets the pixel coordinate of the upper-left corner of the viewport on the render-target surface.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int X { get; set; }
```

Property Value

The upper-left corner of the viewport to set or get.

RemarksUnless rendering is being done only to a subset of the surface, this member can be set to 0.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)

[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

Viewport.Y Property

Gets or sets the pixel coordinate of the upper-left corner of the viewport on the render-target surface.

Namespace: Microsoft.Xna.Framework.Graphics

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Y { get; set; }
```

Property Value

The upper-left corner of the viewport to set or get.

RemarksUnless rendering is being done only to a subset of the surface, this member can be set to 0.

See Also

Reference

[Viewport Structure](#)

[Viewport Members](#)



[Microsoft.Xna.Framework.Graphics Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune




















Microsoft.Xna.Framework.Graphics.PackedVector Namespace

Represents data types with components that are not multiples of 8 bits.

Interfaces

Name	Description
 IPackedVector	Converts packed vector types to and from Vector4 values.
 IPackedVector	Interface that converts packed vector types to and from Vector4 values, allowing multiple encodings to be manipulated in a generic way.

Structures

Name	Description
 Alpha8	Packed vector type containing a single 8 bit normalized W value in the range of 0 to 1.
 Bgr565	Packed vector type containing unsigned normalized values ranging from 0 to 1. The x and z components use 5 bits, and the y component uses 6 bits.
 Bgra5551	Packed vector type containing unsigned normalized values, ranging from 0 to 1, using 5 bits each for x, y, and z, and 1 bit for w.
 Byte4	Packed vector type containing four 8-bit unsigned integer values, ranging from 0 to 255.
 HalfSingle	Packed vector type containing a single 16 bit floating point value.
 HalfVector2	Packed vector type containing two 16-bit floating-point values.
 HalfVector4	Packed vector type containing four 16-bit floating-point values.
 Normalized101010	Packed vector type containing signed normalized values, ranging from -1 to 1, using 10 bits each for x, y, and z, and 2 bits for w.
 NormalizedByte2	Packed vector type containing two 8-bit signed normalized values, ranging from -1 to 1.
 NormalizedByte4	Packed vector type containing four 8-bit signed normalized values, ranging from -1 to 1.
 NormalizedShort2	Packed vector type containing two 16-bit signed normalized values, ranging from -1 to 1.
 NormalizedShort4	Packed vector type containing four 16-bit signed normalized values, ranging from -1 to 1.
 Rg32	Packed vector type containing two 16-bit unsigned normalized values, ranging from 0 to 1.
 Rgba1010102	Packed vector type containing unsigned normalized values, ranging from 0 to 1, using 10 bits each for x, y, and z, and 2 bits for w.
 Rgba32	Packed vector type containing four 8-bit unsigned normalized values, ranging from 0 to 1.
 Rgba64	Packed vector type containing four 16-bit unsigned normalized values, ranging from 0 to 1.
 Short2	Packed vector type containing two 16-bit signed integer values.
 Short4	Packed vector type containing four 16-bit signed integer values.
 UInt101010	Packed vector type containing unsigned integer values, using 10 bits each for x, y, and z (ranging from 0 to 1023), and 2 bits for w (ranging from 0 to 3).

Alpha8 Structure

Packed vector type containing a single 8 bit normalized W value in the range of 0 to 1.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Alpha8 : IPackedVector<byte>, IEquatable<Alpha8>
```

See Also

Reference

[Alpha8 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Alpha8 Members

The following tables list the members exposed by the Alpha8 type.









Public Constructors

Name	Description
 Alpha8	Initializes a new instance of the Alpha8 structure.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToAlpha	Expands the packed representation to a System.Single .
 ToString	Returns a string representation of the current instance.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Alpha8 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Alpha8 Constructor

Initializes a new instance of the [Alpha8](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Alpha8 (  
    float alpha  
)
```

Parameters

alpha

The initial value for the [Alpha8](#) structure.

See Also

Reference

[Alpha8 Structure](#)









[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8 Methods


Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Sop_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Sop_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToAlpha	Expands the packed representation to a System.Single.
	ToString	Returns a string representation of the current instance.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Alpha8 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Alpha8.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Alpha8.Equals (Alpha8)	Returns a value that indicates whether the current instance is equal to a specified object.
Alpha8.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Alpha8.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Alpha8.Equals Method (Alpha8)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Alpha8 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representaion from.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Alpha8 a,  
    Alpha8 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Alpha8 a,  
    Alpha8 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8.ToAlpha Method

Expands the packed representation to a [System.Single](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float ToAlpha ()
```

Return Value

The expanded [Alpha8](#).

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Alpha8 Structure](#)


[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Alpha8 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Alpha8 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Alpha8.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Alpha8 Structure](#)

[Alpha8 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565 Structure

Packed vector type containing unsigned normalized values ranging from 0 to 1. The x and z components use 5 bits, and the y component uses 6 bits.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Bgr565 : IPackedVector<UInt16>, IEquatable<Bgr565>
```

See Also

Reference

[Bgr565 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Bgr565 Members

The following tables list the members exposed by the Bgr565 type.









Public Constructors

Name	Description
 Bgr565	Overloaded. Initializes a new instance of the Bgr565 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector3	Expands the packed representation into a Vector3 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Bgr565 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgr565 Constructor

Initializes a new instance of the [Bgr565](#) class.

Overload List

Name	Description
Bgr565 (Single, Single, Single)	Initializes a new instance of the Bgr565 class.
Bgr565 (Vector3)	Initializes a new instance of the Bgr565 structure.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgr565 Constructor (Single, Single, Single)

Initializes a new instance of the [Bgr565](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Bgr565 (  
    float x,  
    float y,  
    float z  
)
```

Parameters

x
Initial value for the x component.

y
Initial value for the y component.

z
Initial value for the z component.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565 Constructor (Vector3)

Initializes a new instance of the [Bgr565](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Bgr565 (  
    Vector3 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Bgr565](#) structure.

See Also

Reference

[Bgr565 Structure](#)











[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565 Methods


Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector3	Expands the packed representation into a Vector3 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Bgr565 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgr565.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Bgr565.Equals (Bgr565)	Returns a value that indicates whether the current instance is equal to a specified object.
Bgr565.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Bgr565.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgr565.Equals Method (Bgr565)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Bgr565 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Bgr565 a,  
    Bgr565 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Bgr565 a,  
    Bgr565 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565.ToVector3 Method

Expands the packed representation into a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 ToVector3 ()
```

Return Value

The expanded vector.

See Also

Reference

[Bgr565 Structure](#)


[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgr565 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Bgr565 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgr565.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public UInt16 PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Bgr565 Structure](#)

[Bgr565 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551 Structure

Packed vector type containing unsigned normalized values, ranging from 0 to 1, using 5 bits each for x, y, and z, and 1 bit for w.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Bgra5551 : IPackedVector<UInt16>, IEquatable<Bgra5551>
```

See Also

Reference

[Bgra5551 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Bgra5551 Members

The following tables list the members exposed by the Bgra5551 type.









Public Constructors

	Name	Description
	Bgra5551	Overloaded. Initializes a new instance of the Bgra5551 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Sop_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Sop_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Bgra5551 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgra5551 Constructor

Initializes a new instance of the [Bgra5551](#) class.

Overload List

Name	Description
Bgra5551 (Single, Single, Single, Single)	Initializes a new instance of the Bgra5551 class.
Bgra5551 (Vector4)	Initializes a new instance of the Bgra5551 structure.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgra5551 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [Bgra5551](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Bgra5551 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

- x*
Initial value for the x component.
- y*
Initial value for the y component.
- z*
Initial value for the z component.
- w*
Initial value for the w component.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551 Constructor (Vector4)

Initializes a new instance of the [Bgra5551](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Bgra5551 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Bgra5551](#) structure.

See Also

Reference

[Bgra5551 Structure](#)









[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Bgra5551 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgra5551.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Bgra5551.Equals (Bgra5551)	Returns a value that indicates whether the current instance is equal to a specified object.
Bgra5551.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Bgra5551.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgra5551.Equals Method (Bgra5551)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Bgra5551 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Bgra5551 a,  
    Bgra5551 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Bgra5551 a,  
    Bgra5551 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Bgra5551 Structure](#)


[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Bgra5551 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Bgra5551 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Bgra5551.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public UInt16 PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Bgra5551 Structure](#)

[Bgra5551 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4 Structure

Packed vector type containing four 8-bit unsigned integer values, ranging from 0 to 255.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Byte4 : IPackedVector<uint>, IEquatable<Byte4>
```

See Also

Reference

[Byte4 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Byte4 Members

The following tables list the members exposed by the Byte4 type.











Public Constructors

	Name	Description
	Byte4	Overloaded. Initializes a new instance of the Byte4 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4

See Also

Reference

[Byte4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Byte4 Constructor

Initializes a new instance of the [Byte4](#) class.

Overload List

Name	Description
Byte4 (Single, Single, Single, Single)	Initializes a new instance of the Byte4 class.
Byte4 (Vector4)	Initializes a new instance of the Byte4 structure.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Byte4 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [Byte4](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Byte4 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4 Constructor (Vector4)

Initializes a new instance of the [Byte4](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Byte4 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Byte4](#) structure.

See Also

Reference

[Byte4 Structure](#)









[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4

See Also

Reference

[Byte4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Byte4.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Byte4.Equals (Byte4)	Returns a value that indicates whether the current instance is equal to a specified object.
Byte4.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Byte4.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Byte4.Equals Method (Byte4)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Byte4 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#)

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create packed representation from.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Byte4 a,  
    Byte4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Byte4 a,  
    Byte4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Byte4 Structure](#)


[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Byte4 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Byte4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Byte4.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Byte4 Structure](#)

[Byte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle Structure

Packed vector type containing a single 16 bit floating point value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct HalfSingle : IPackedVector<UInt16>, IEquatable<HalfSingle>
```

See Also

Reference

[HalfSingle Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


HalfSingle Members

The following tables list the members exposed by the HalfSingle type.









Public Constructors

Name	Description
 HalfSingle	Initializes a new instance of the HalfSingle structure.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.


Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Sop_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Sop_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToSingle	Expands the HalfSingle to a Single.
 ToString	Returns a string representation of the current instance.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[HalfSingle Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfSingle Constructor

Initializes a new instance of the [HalfSingle](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public HalfSingle (  
    float value  
)
```

Parameters

value

The initial value of the [HalfSingle](#) structure.

See Also

Reference

[HalfSingle Structure](#)











[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle Methods



Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToSingle	Expands the HalfSingle to a Single.
	ToString	Returns a string representation of the current instance.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[HalfSingle Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfSingle.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
HalfSingle.Equals (HalfSingle)	Returns a value that indicates whether the current instance is equal to a specified object.
HalfSingle.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
HalfSingle.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfSingle.Equals Method (HalfSingle)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    HalfSingle other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    HalfSingle a,  
    HalfSingle b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    HalfSingle a,  
    HalfSingle b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle.ToSingle Method

Expands the [HalfSingle](#) to a [Single](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float ToSingle ()
```

Return Value

The expanded [HalfSingle](#).

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[HalfSingle Structure](#)


[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfSingle Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[HalfSingle Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfSingle.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public UInt16 PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[HalfSingle Structure](#)

[HalfSingle Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2 Structure

Packed vector type containing two 16-bit floating-point values.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct HalfVector2 : IPackedVector<uint>, IEquatable<HalfVector2>
```

See Also

Reference

[HalfVector2 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


HalfVector2 Members

The following tables list the members exposed by the HalfVector2 type.









Public Constructors

Name	Description
 HalfVector2	Overloaded. Initializes a new instance of the HalfVector2 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector2	Expands the HalfVector2 to a Vector2 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[HalfVector2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector2 Constructor

Initializes a new instance of the [HalfVector2](#) class.

Overload List

Name	Description
HalfVector2 (Single, Single)	Initializes a new instance of the HalfVector2 structure.
HalfVector2 (Vector2)	Initializes a new instance of the HalfVector2 structure.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector2 Constructor (Single, Single)

Initializes a new instance of the [HalfVector2](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public HalfVector2 (  
    float x,  
    float y  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2 Constructor (Vector2)

Initializes a new instance of the [HalfVector2](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public HalfVector2 (  
    Vector2 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [HalfVector2](#) structure.

See Also

Reference

[HalfVector2 Structure](#)











[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the HalfVector2 to a Vector2 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[HalfVector2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector2.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
HalfVector2.Equals (HalfVector2)	Returns a value that indicates whether the current instance is equal to a specified object.
HalfVector2.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
HalfVector2.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector2.Equals Method (HalfVector2)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    HalfVector2 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    HalfVector2 a,  
    HalfVector2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    HalfVector2 a,  
    HalfVector2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2.ToVector2 Method

Expands the [HalfVector2](#) to a [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ToVector2 ()
```

Return Value

The expanded [HalfVector2](#).

See Also

Reference

[HalfVector2 Structure](#)


[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector2 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[HalfVector2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector2.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[HalfVector2 Structure](#)

[HalfVector2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4 Structure

Packed vector type containing four 16-bit floating-point values.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct HalfVector4 : IPackedVector<ulong>, IEquatable<HalfVector4>
```

See Also

Reference

[HalfVector4 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


HalfVector4 Members

The following tables list the members exposed by the HalfVector4 type.









Public Constructors

Name	Description
 HalfVector4	Overloaded. Initializes a new instance of the HalfVector4 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[HalfVector4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector4 Constructor

Initializes a new instance of the [HalfVector4](#) class.

Overload List

Name	Description
HalfVector4 (Single, Single, Single, Single)	Initializes a new instance of the HalfVector4 class.
HalfVector4 (Vector4)	Initializes a new instance of the HalfVector4 structure.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector4 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [HalfVector4](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public HalfVector4 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4 Constructor (Vector4)

Initializes a new instance of the [HalfVector4](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public HalfVector4 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [HalfVector4](#) structure.

See Also

Reference

[HalfVector4 Structure](#)









[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[HalfVector4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector4.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
HalfVector4.Equals (HalfVector4)	Returns a value that indicates whether the current instance is equal to a specified object.
HalfVector4.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
HalfVector4.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector4.Equals Method (HalfVector4)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    HalfVector4 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    HalfVector4 a,  
    HalfVector4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    HalfVector4 a,  
    HalfVector4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[HalfVector4 Structure](#)


[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HalfVector4 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[HalfVector4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

HalfVector4.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public ulong PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[HalfVector4 Structure](#)

[HalfVector4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IPackedVector Generic Interface

Converts packed vector types to and from [Vector4](#) values.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public interface IPackedVector<TPacked> : IPackedVector
```

See Also

Reference

[IPackedVector Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IPackedVector Members

The following tables list the members exposed by the IPackedVector type.

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also


Reference

[IPackedVector Generic Interface](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

IPackedVector Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[IPackedVector Generic Interface](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

IPackedVector.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public abstract TPacked PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[IPackedVector Generic Interface](#)

[IPackedVector Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IPackedVector Interface

Interface that converts packed vector types to and from [Vector4](#) values, allowing multiple encodings to be manipulated in a generic way.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public interface IPackedVector
```

See Also

Reference

[IPackedVector Members](#)



[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IPackedVector Members

The following tables list the members exposed by the IPackedVector type.

Public Methods

	Name	Description
	PackFromVector4	Sets the packed representation from a Vector4 .
	ToVector4	Expands the packed representation into a Vector4 .

See Also



Reference

[IPackedVector Interface](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

IPackedVector Methods

Public Methods

	Name	Description
	PackFromVector4	Sets the packed representation from a Vector4 .
	ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[IPackedVector Interface](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void PackFromVector4 (  
    Vector4 vector  
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[IPackedVector Interface](#)

[IPackedVector Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[IPackedVector Interface](#)

[IPackedVector Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010 Structure

Packed vector type containing signed normalized values, ranging from -1 to 1 , using 10 bits each for x, y, and z, and 2 bits for w.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Normalized101010 : IPackedVector<uint>, IEquatable<Normalized101010>
```

See Also

Reference

[Normalized101010 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Normalized101010 Members

The following tables list the members exposed by the Normalized101010 type.









Public Constructors

Name	Description
 Normalized101010	Overloaded. Initializes a new instance of the Normalized101010 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector3	Expands the packed representation into a Vector3 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Normalized101010 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Normalized101010 Constructor

Initializes a new instance of the [Normalized101010](#) class.

Overload List

Name	Description
Normalized101010 (Single, Single, Single)	Initializes a new instance of the Normalized101010 class.
Normalized101010 (Vector3)	Initializes a new instance of the Normalized101010 structure.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Normalized101010 Constructor (Single, Single, Single)

Initializes a new instance of the [Normalized101010](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Normalized101010 (  
    float x,  
    float y,  
    float z  
)
```

Parameters

x
Initial value for the x component.

y
Initial value for the y component.

z
Initial value for the z component.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010 Constructor (Vector3)

Initializes a new instance of the [Normalized101010](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Normalized101010 (  
    Vector3 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Normalized101010](#) structure.

See Also

Reference

[Normalized101010 Structure](#)











[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector3	Expands the packed representation into a Vector3 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Normalized101010 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Normalized101010.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Normalized101010.Equals (Normalized101010)	Returns a value that indicates whether the current instance is equal to a specified object.
Normalized101010.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Normalized101010.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Normalized101010.Equals Method (Normalized101010)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Normalized101010 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Normalized101010 a,  
    Normalized101010 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Normalized101010 a,  
    Normalized101010 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010.ToVector3 Method

Expands the packed representation into a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 ToVector3 ()
```

Return Value

The expanded vector.

See Also

Reference

[Normalized101010 Structure](#)


[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Normalized101010 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Normalized101010 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Normalized101010.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Normalized101010 Structure](#)

[Normalized101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2 Structure

Packed vector type containing two 8-bit signed normalized values, ranging from -1 to 1.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct NormalizedByte2 : IPackedVector<UInt16>, IEquatable<NormalizedByte2>
```

See Also

Reference

[NormalizedByte2 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


NormalizedByte2 Members

The following tables list the members exposed by the NormalizedByte2 type.









Public Constructors

Name	Description
 NormalizedByte2	Overloaded. Initializes a new instance of the NormalizedByte2 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector2	Expands the packed representation to a vector.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[NormalizedByte2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte2 Constructor

Initializes a new instance of the [NormalizedByte2](#) class.

Overload List

Name	Description
NormalizedByte2 (Single, Single)	Initializes a new instance of the NormalizedByte2 class.
NormalizedByte2 (Vector2)	Initializes a new instance of the NormalizedByte2 structure.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte2 Constructor (Single, Single)

Initializes a new instance of the [NormalizedByte2](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedByte2 (  
    float x,  
    float y  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2 Constructor (Vector2)

Initializes a new instance of the [NormalizedByte2](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedByte2 (  
    Vector2 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [NormalizedByte2](#) structure.

See Also

Reference

[NormalizedByte2 Structure](#)











[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
 	op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 	op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the packed representation to a vector.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[NormalizedByte2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte2.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
NormalizedByte2.Equals (NormalizedByte2)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedByte2.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedByte2.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte2.Equals Method (NormalizedByte2)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    NormalizedByte2 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    NormalizedByte2 a,  
    NormalizedByte2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    NormalizedByte2 a,  
    NormalizedByte2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2.ToVector2 Method

Expands the packed representation to a vector.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ToVector2 ()
```

Return Value

The expanded [NormalizedByte2](#).

See Also

Reference

[NormalizedByte2 Structure](#)


[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte2 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[NormalizedByte2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte2.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public UInt16 PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[NormalizedByte2 Structure](#)

[NormalizedByte2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4 Structure

Packed vector type containing four 8-bit signed normalized values, ranging from -1 to 1 .

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

```
C#  
  
public struct NormalizedByte4 : IPackedVector<uint>, IEquatable<NormalizedByte4>
```

See Also

Reference

[NormalizedByte4 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


NormalizedByte4 Members

The following tables list the members exposed by the NormalizedByte4 type.









Public Constructors

Name	Description
 NormalizedByte4	Overloaded. Initializes a new instance of the NormalizedByte4 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[NormalizedByte4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte4 Constructor

Initializes a new instance of the [NormalizedByte4](#) class.

Overload List

Name	Description
NormalizedByte4 (Single, Single, Single, Single)	Initializes a new instance of the NormalizedByte4 class.
NormalizedByte4 (Vector4)	Initializes a new instance of the NormalizedByte4 structure.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte4 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [NormalizedByte4](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedByte4 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4 Constructor (Vector4)

Initializes a new instance of the [NormalizedByte4](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedByte4 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [NormalizedByte4](#) structure.

See Also

Reference

[NormalizedByte4 Structure](#)









[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[NormalizedByte4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte4.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
NormalizedByte4.Equals (NormalizedByte4)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedByte4.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedByte4.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte4.Equals Method (NormalizedByte4)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    NormalizedByte4 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    NormalizedByte4 a,  
    NormalizedByte4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    NormalizedByte4 a,  
    NormalizedByte4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[NormalizedByte4 Structure](#)


[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedByte4 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[NormalizedByte4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedByte4.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[NormalizedByte4 Structure](#)

[NormalizedByte4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2 Structure

Packed vector type containing two 16-bit signed normalized values, ranging from -1 to 1.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct NormalizedShort2 : IPackedVector<uint>, IEquatable<NormalizedShort2>
```

See Also

Reference

[NormalizedShort2 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


NormalizedShort2 Members

The following tables list the members exposed by the NormalizedShort2 type.









Public Constructors

Name	Description
 NormalizedShort2	Overloaded. Initializes a new instance of the NormalizedShort2 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector2	Expands the packed representation to a vector.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[NormalizedShort2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort2 Constructor

Initializes a new instance of the [NormalizedShort2](#) class.

Overload List

Name	Description
NormalizedShort2 (Single, Single)	Initializes a new instance of the NormalizedShort2 class.
NormalizedShort2 (Vector2)	Initializes a new instance of the NormalizedShort2 structure.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort2 Constructor (Single, Single)

Initializes a new instance of the [NormalizedShort2](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedShort2 (  
    float x,  
    float y  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2 Constructor (Vector2)

Initializes a new instance of the [NormalizedShort2](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedShort2 (  
    Vector2 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [NormalizedShort2](#) structure.

See Also

Reference

[NormalizedShort2 Structure](#)











[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2 Methods



Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the packed representation to a vector.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[NormalizedShort2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort2.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
NormalizedShort2.Equals (NormalizedShort2)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedShort2.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedShort2.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort2.Equals Method (NormalizedShort2)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    NormalizedShort2 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    NormalizedShort2 a,  
    NormalizedShort2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    NormalizedShort2 a,  
    NormalizedShort2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2.ToVector2 Method

Expands the packed representation to a vector.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ToVector2 ()
```

Return Value

The expanded [NormalizedShort2](#).

See Also

Reference

[NormalizedShort2 Structure](#)


[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort2 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[NormalizedShort2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort2.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[NormalizedShort2 Structure](#)

[NormalizedShort2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4 Structure

Packed vector type containing four 16-bit signed normalized values, ranging from -1 to 1 .

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct NormalizedShort4 : IPackedVector<ulong>, IEquatable<NormalizedShort4>
```

See Also

Reference

[NormalizedShort4 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


NormalizedShort4 Members

The following tables list the members exposed by the NormalizedShort4 type.









Public Constructors

Name	Description
 NormalizedShort4	Overloaded. Initializes a new instance of the NormalizedShort4 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[NormalizedShort4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort4 Constructor

Initializes a new instance of the [NormalizedShort4](#) class.

Overload List

Name	Description
NormalizedShort4 (Single, Single, Single, Single)	Initializes a new instance of the NormalizedShort4 class.
NormalizedShort4 (Vector4)	Initializes a new instance of the NormalizedShort4 structure.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort4 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [NormalizedShort4](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedShort4 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4 Constructor (Vector4)

Initializes a new instance of the [NormalizedShort4](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NormalizedShort4 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [NormalizedShort4](#) structure.

See Also

Reference

[NormalizedShort4 Structure](#)











[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[NormalizedShort4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort4.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
NormalizedShort4.Equals (NormalizedShort4)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedShort4.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
NormalizedShort4.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort4.Equals Method (NormalizedShort4)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    NormalizedShort4 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    NormalizedShort4 a,  
    NormalizedShort4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    NormalizedShort4 a,  
    NormalizedShort4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[NormalizedShort4 Structure](#)


[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NormalizedShort4 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[NormalizedShort4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

NormalizedShort4.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public ulong PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[NormalizedShort4 Structure](#)

[NormalizedShort4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32 Structure

Packed vector type containing two 16-bit unsigned normalized values, ranging from 0 to 1.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Rg32 : IPackedVector<uint>, IEquatable<Rg32>
```

See Also

Reference

[Rg32 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Rg32 Members

The following tables list the members exposed by the Rg32 type.









Public Constructors

	Name	Description
	Rg32	Overloaded. Initializes a new instance of the Rg32 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the packed vector representation into a Vector2 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Rg32 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rg32 Constructor

Initializes a new instance of the [Rg32](#) class.

Overload List

Name	Description
Rg32 (Single, Single)	Initializes a new instance of the Rg32 structure.
Rg32 (Vector2)	Initializes a new instance of the Rg32 structure.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rg32 Constructor (Single, Single)

Initializes a new instance of the [Rg32](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rg32 (
    float x,
    float y
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32 Constructor (Vector2)

Initializes a new instance of the [Rg32](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rg32 (  
    Vector2 vector  
)
```

Parameters

vector

The vector containing the initial values for the components of the [Rg32](#) structure.

See Also

Reference

[Rg32 Structure](#)











[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
 	op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 	op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the packed vector representation into a Vector2 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Rg32 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rg32.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Rg32.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Rg32.Equals (Rg32)	Returns a value that indicates whether the current instance is equal to a specified object.
Rg32.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rg32.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32.Equals Method (Rg32)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Rg32 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#)

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create packed representation from.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Rg32 a,  
    Rg32 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Rg32 a,  
    Rg32 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32.ToVector2 Method

Expands the packed vector representation into a [Vector2](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ToVector2 ()
```

Return Value

The expanded vector.

See Also

Reference

[Rg32 Structure](#)


[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rg32 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Rg32 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rg32.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Rg32 Structure](#)

[Rg32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102 Structure

Packed vector type containing unsigned normalized values, ranging from 0 to 1, using 10 bits each for x, y, and z, and 2 bits for w.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Rgba1010102 : IPackedVector<uint>, IEquatable<Rgba1010102>
```

See Also

Reference

[Rgba1010102 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Rgba1010102 Members

The following tables list the members exposed by the Rgba1010102 type.









Public Constructors

Name	Description
 Rgba1010102	Overloaded. Initializes a new instance of the Rgba1010102 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4

See Also

Reference

[Rgba1010102 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba1010102 Constructor

Initializes a new instance of the [Rgba1010102](#) class.

Overload List

Name	Description
Rgba1010102 (Single, Single, Single, Single)	Initializes a new instance of the Rgba1010102 class.
Rgba1010102 (Vector4)	Initializes a new instance of the Rgba1010102 structure.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba1010102 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [Rgba1010102](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rgba1010102 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

- x*
Initial value for the x component.
- y*
Initial value for the y component.
- z*
Initial value for the z component.
- w*
Initial value for the w component.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102 Constructor (Vector4)

Initializes a new instance of the [Rgba1010102](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rgba1010102 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Rgba1010102](#) structure.

See Also

Reference

[Rgba1010102 Structure](#)











[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4

See Also

Reference

[Rgba1010102 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba1010102.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Rgba1010102.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Rgba1010102.Equals (Rgba1010102)	Returns a value that indicates whether the current instance is equal to a specified object.
Rgba1010102.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba1010102.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102.Equals Method (Rgba1010102)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Rgba1010102 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#)

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Rgba1010102 a,  
    Rgba1010102 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the left of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Rgba1010102 a,  
    Rgba1010102 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Rgba1010102 Structure](#)


[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba1010102 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Rgba1010102 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba1010102.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Rgba1010102 Structure](#)

[Rgba1010102 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32 Structure

Packed vector type containing four 8-bit unsigned normalized values, ranging from 0 to 1.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Rgba32 : IPackedVector<uint>, IEquatable<Rgba32>
```

See Also

Reference

[Rgba32 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Rgba32 Members

The following tables list the members exposed by the Rgba32 type.









Public Constructors

	Name	Description
	Rgba32	Overloaded. Initializes a new instance of the Rgba32 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Rgba32 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba32 Constructor

Initializes a new instance of the [Rgba32](#) class.

Overload List

Name	Description
Rgba32 (Single, Single, Single, Single)	Initializes a new instance of the Rgba32 class.
Rgba32 (Vector4)	Initializes a new instance of the Rgba32 structure.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba32 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [Rgba32](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rgba32 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32 Constructor (Vector4)

Initializes a new instance of the [Rgba32](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rgba32 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Rgba32](#) structure.

See Also

Reference

[Rgba32 Structure](#)











[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32 Methods


Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Rgba32 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba32.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Rgba32.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Rgba32.Equals (Rgba32)	Returns a value that indicates whether the current instance is equal to a specified object.
Rgba32.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba32.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32.Equals Method (Rgba32)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Rgba32 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Rgba32 a,  
    Rgba32 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Rgba32 a,  
    Rgba32 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Rgba32 Structure](#)


[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba32 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Rgba32 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba32.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Rgba32 Structure](#)

[Rgba32 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64 Structure

Packed vector type containing four 16-bit unsigned normalized values, ranging from 0 to 1.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Rgba64 : IPackedVector<ulong>, IEquatable<Rgba64>
```

See Also

Reference

[Rgba64 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Rgba64 Members

The following tables list the members exposed by the Rgba64 type.









Public Constructors

	Name	Description
	Rgba64	Overloaded. Initializes a new instance of the Rgba64 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Rgba64 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba64 Constructor

Initializes a new instance of the [Rgba64](#) class.

Overload List

Name	Description
Rgba64 (Single, Single, Single, Single)	Initializes a new instance of the Rgba64 structure.
Rgba64 (Vector4)	Initializes a new instance of the Rgba64 structure.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba64 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [Rgba64](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rgba64 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64 Constructor (Vector4)

Initializes a new instance of the [Rgba64](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Rgba64 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Rgba64](#) structure.

See Also

Reference

[Rgba64 Structure](#)











[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Rgba64 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba64.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Rgba64.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Rgba64.Equals (Rgba64)	Returns a value that indicates whether the current instance is equal to a specified object.
Rgba64.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba64.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64.Equals Method (Rgba64)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Rgba64 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Rgba64 a,  
    Rgba64 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Rgba64 a,  
    Rgba64 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Rgba64 Structure](#)


[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Rgba64 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Rgba64 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Rgba64.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public ulong PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Rgba64 Structure](#)

[Rgba64 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2 Structure

Packed vector type containing two 16-bit signed integer values.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Short2 : IPackedVector<uint>, IEquatable<Short2>
```

See Also

Reference

[Short2 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Short2 Members

The following tables list the members exposed by the Short2 type.









Public Constructors

	Name	Description
	Short2	Overloaded. Initializes a new instance of the Short2 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.


Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the packed representation to a vector.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Short2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short2 Constructor

Initializes a new instance of the [Short2](#) class.

Overload List

Name	Description
Short2 (Single, Single)	Initializes a new instance of the Short2 class.
Short2 (Vector2)	Initializes a new instance of the Short2 structure.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short2 Constructor (Single, Single)

Initializes a new instance of the [Short2](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Short2 (  
    float x,  
    float y  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2 Constructor (Vector2)

Initializes a new instance of the [Short2](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Short2 (  
    Vector2 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Short2](#) structure.

See Also

Reference

[Short2 Structure](#)











[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2 Methods



Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector2	Expands the packed representation to a vector.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[Short2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short2.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Short2.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Short2.Equals (Short2)	Returns a value that indicates whether the current instance is equal to a specified object.
Short2.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short2.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2.Equals Method (Short2)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Short2 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Short2 a,  
    Short2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Short2 a,  
    Short2 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2.ToVector2 Method

Expands the packed representation to a vector.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ToVector2 ()
```

Return Value

The expanded [Short2](#).

See Also

Reference

[Short2 Structure](#)


[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short2 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Short2 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short2.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Short2 Structure](#)

[Short2 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4 Structure

Packed vector type containing four 16-bit signed integer values.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct Short4 : IPackedVector<ulong>, IEquatable<Short4>
```

See Also

Reference

[Short4 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


Short4 Members

The following tables list the members exposed by the Short4 type.









Public Constructors

	Name	Description
	Short4	Overloaded. Initializes a new instance of the Short4 class.



Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Short4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short4 Constructor

Initializes a new instance of the [Short4](#) class.

Overload List

Name	Description
Short4 (Single, Single, Single, Single)	Initializes a new instance of the Short4 class.
Short4 (Vector4)	Initializes a new instance of the Short4 structure.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short4 Constructor (Single, Single, Single, Single)

Initializes a new instance of the [Short4](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Short4 (  
    float x,  
    float y,  
    float z,  
    float w  
)
```

Parameters

x

Initial value for the x component.

y

Initial value for the y component.

z

Initial value for the z component.

w

Initial value for the w component.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4 Constructor (Vector4)

Initializes a new instance of the [Short4](#) structure.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Short4 (  
    Vector4 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [Short4](#) structure.

See Also

Reference

[Short4 Structure](#)











[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4 Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
	 op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
	 op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector4	Expands the packed representation into a Vector4 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .

See Also

Reference

[Short4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short4.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
Short4.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
Short4.Equals (Short4)	Returns a value that indicates whether the current instance is equal to a specified object.
Short4.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short4.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4.Equals Method (Short4)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Short4 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Short4 a,  
    Short4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Short4 a,  
    Short4 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[Short4 Structure](#)


[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Short4 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[Short4 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Short4.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public ulong PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[Short4 Structure](#)

[Short4 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010 Structure

Packed vector type containing unsigned integer values, using 10 bits each for x, y, and z (ranging from 0 to 1023), and 2 bits for w (ranging from 0 to 3).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct UInt101010 : IPackedVector<uint>, IEquatable<UInt101010>
```

See Also

Reference

[UInt101010 Members](#)


[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


UInt101010 Members

The following tables list the members exposed by the UInt101010 type.









Public Constructors

Name	Description
 UInt101010	Overloaded. Initializes a new instance of the UInt101010 class.



Public Properties

Name	Description
 PackedValue	Directly gets or sets the packed representation of the value.



Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for the current instance.
 GetType	(Inherited from Object .)
 Op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 Op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a string representation of the current instance.
 ToVector3	Expands the packed representation into a Vector3 .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[UInt101010 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

UInt101010 Constructor

Initializes a new instance of the [UInt101010](#) class.

Overload List

Name	Description
UInt101010 (Single, Single, Single)	Initializes a new instance of the UInt101010 class.
UInt101010 (Vector3)	Initializes a new instance of the UInt101010 class.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

UInt101010 Constructor (Single, Single, Single)

Initializes a new instance of the [UInt101010](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public UInt101010 (  
    float x,  
    float y,  
    float z  
)
```

Parameters

x
Initial value for the x component.

y
Initial value for the y component.

z
Initial value for the z component.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010 Constructor (Vector3)

Initializes a new instance of the [UInt101010](#) class.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public UInt101010 (  
    Vector3 vector  
)
```

Parameters

vector

A vector containing the initial values for the components of the [UInt101010](#) structure.

See Also

Reference

[UInt101010 Structure](#)











[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010 Methods


Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for the current instance.
	GetType	(Inherited from Object .)
 	op_Equality	Compares the current instance of a class to another instance to determine whether they are the same.
 	op_Inequality	Compares the current instance of a class to another instance to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a string representation of the current instance.
	ToVector3	Expands the packed representation into a Vector3 .

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4	Sets the packed representation from a Vector4 .
	Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4	Expands the packed representation into a Vector4 .

See Also

Reference

[UInt101010 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

UInt101010.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
UInt101010.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
UInt101010.Equals (UInt101010)	Returns a value that indicates whether the current instance is equal to a specified object.
UInt101010.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

UInt101010.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010.Equals Method (UInt101010)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    UInt101010 other  
)
```

Parameters

other

The object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010.GetHashCode Method

Gets the hash code for the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for the instance.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 Method

Sets the packed representation from a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.PackFromVector4 (
    Vector4 vector
)
```

Parameters

vector

The vector to create the packed representation from.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 Method

Expands the packed representation into a [Vector4](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Vector4 Microsoft.Xna.Framework.Graphics.PackedVector.IPackedVector.ToVector4 ()
```

Return Value

The expanded vector.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010.op_Equality Method

Compares the current instance of a class to another instance to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    UInt101010 a,  
    UInt101010 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010.op_Inequality Method

Compares the current instance of a class to another instance to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    UInt101010 a,  
    UInt101010 b  
)
```

Parameters

a

The object to the left of the equality operator.

b

The object to the right of the equality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010.ToString Method

Returns a string representation of the current instance.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String that represents the object.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010.ToVector3 Method

Expands the packed representation into a [Vector3](#).

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 ToVector3 ()
```

Return Value

The expanded vector.

See Also

Reference

[UInt101010 Structure](#)


[UInt101010 Members](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

UInt101010 Properties

Public Properties

	Name	Description
	PackedValue	Directly gets or sets the packed representation of the value.

See Also

Reference

[UInt101010 Structure](#)

[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

UInt101010.PackedValue Property

Directly gets or sets the packed representation of the value.

Namespace: Microsoft.Xna.Framework.Graphics.PackedVector

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[CLSCompliantAttribute(false)]  
public uint PackedValue { get; set; }
```

Property Value

The packed representation of the value.

See Also

Reference

[UInt101010 Structure](#)

[UInt101010 Members](#)






[Microsoft.Xna.Framework.Graphics.PackedVector Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune















Microsoft.Xna.Framework.Input Namespace

Contains classes to receive input from keyboard, mouse, and Xbox 360 Controller devices.








Classes

Name	Description
 Accelerometer	Provides methods for interacting with the 3-axis accelerometer of a Zune device.
 GamePad	Allows retrieval of user interaction with an Xbox 360 Controller and setting of controller vibration motors.
 Keyboard	Allows retrieval of keystrokes from a keyboard input device.
 Mouse	Allows retrieval of position and button clicks from a mouse input device.
 TouchPanel	Provides methods for retrieving touch panel device information.

Structures

Name	Description
 AccelerometerCapabilities	Provides properties for accessing the capabilities of an accelerometer.
 AccelerometerState	Provides information on the current state of the accelerometer device and a helper function for rotational computation.
 GamePadButtons	Identifies whether the buttons on an Xbox 360 Controller are pressed or released.
 GamePadCapabilities	Describes the capabilities of an Xbox 360 Controller, including controller type and whether the controller supports voice.
 GamePadDPad	Identifies which directions on the directional pad of an Xbox 360 Controller are being pressed.
 GamePadState	Represents specific information about the state of an Xbox 360 Controller, including the current state of buttons and sticks.
 GamePadThumbSticks	Structure that represents the position of left and right sticks (thumbsticks) on an Xbox 360 Controller.
 GamePadTriggers	Structure that defines the position of the left and right triggers on an Xbox 360 controller.
 KeyboardState	Represents a state of keystrokes recorded by a keyboard input device.
 MouseState	Represents the state of a mouse input device, including mouse cursor position and buttons pressed.
 TouchCollection	Provides methods and properties for accessing state information for the touch screen of a Zune device.
 TouchCollection.Enumerator	Provides the ability to iterate through the touch locations in a TouchCollection .
 TouchLocation	Provides methods and properties for interacting with a touch location on a touch screen device.
 TouchPanelCapabilities	Provides access to information about the touch pad device.

Enumerations

Name	Description
 Buttons	Enumerates input device buttons.
 ButtonState	Identifies the state of a controller button.
 GamePadDeadZone	Specifies a type of dead zone processing to apply to Xbox 360 controller analog sticks when calling GetState .
 GamePadType	Describes the type of a specified Xbox 360 controller.
 Keys	Identifies a particular key on a keyboard.
 KeyState	Identifies the state of a keyboard key.
 TouchLocationState	Defines the possible states for a touch location.

See Also

Tasks

- [How To: Detect Whether a Controller Button Is Pressed](#)
- [How To: Detect Whether a Controller Is Disconnected](#)
- [How To: Get the Current Mouse Position \(Windows\)](#)
- [How To: Detect Whether a Key Is Pressed](#)

Concepts

- [Input Overview](#)
- [Input Content Catalog at XNA Creators Club Online](#)

Accelerometer Class

Note

This class is available only when developing for Zune.

Provides methods for interacting with the 3-axis accelerometer of a Zune device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class Accelerometer
```

See Also

Reference

[Accelerometer Members](#)










[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune



Accelerometer Members

The following tables list the members exposed by the Accelerometer type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
 	GetCapabilities	Get the accelerometer capabilities of this device.
	GetHashCode	(Inherited from Object .)
 	GetState	Gets the current state of the accelerometer.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also










Reference

[Accelerometer Class](#)



[Microsoft.Xna.Framework.Input Namespace](#)

Accelerometer Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
 	GetCapabilities	Get the accelerometer capabilities of this device.
	GetHashCode	(Inherited from Object .)
 	GetState	Gets the current state of the accelerometer.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Accelerometer Class](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Accelerometer.GetCapabilities Method

Note

This method is available only when developing for Zune.

Get the accelerometer capabilities of this device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AccelerometerCapabilities GetCapabilities ()
```

Return Value

Capabilities of the accelerometer device.

See Also

Reference

[Accelerometer Class](#)

[Accelerometer Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

Accelerometer.GetState Method

Note

This method is available only when developing for Zune.

Gets the current state of the accelerometer.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AccelerometerState GetState ()
```

Return Value

State of the device's accelerometer.

See Also

Reference

[Accelerometer Class](#)

[Accelerometer Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerCapabilities Structure

Note

This structure is available only when developing for Zune.

Provides properties for accessing the capabilities of an accelerometer.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct AccelerometerCapabilities
```

See Also

Reference

[AccelerometerCapabilities Members](#)








[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune




AccelerometerCapabilities Members

The following tables list the members exposed by the AccelerometerCapabilities type.



Public Properties

Name	Description
 AccelerationResolution	Gets the acceleration resolution, in G-force (g), of the accelerometer.
 HasXAxis	Indicates if the accelerometer has an X-axis.
 HasYAxis	Indicates if the accelerometer has an Y-axis.
 HasZAxis	Indicates if the accelerometer has an Z-axis.
 IsConnected	Indicates if an accelerometer device is available for use.
 MaximumAcceleration	Gets the maximum acceleration value, in G-force (g), of the accelerometer.
 MinimumAcceleration	Gets the minimum acceleration value, in G-force (g), of the accelerometer.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also




Reference

[AccelerometerCapabilities Structure](#)



[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerCapabilities Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[AccelerometerCapabilities Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerCapabilities Properties

Public Properties

	Name	Description
	AccelerationResolution	Gets the acceleration resolution, in G-force (g), of the accelerometer.
	HasXAxis	Indicates if the accelerometer has an X-axis.
	HasYAxis	Indicates if the accelerometer has an Y-axis.
	HasZAxis	Indicates if the accelerometer has an Z-axis.
	IsConnected	Indicates if an accelerometer device is available for use.
	MaximumAcceleration	Gets the maximum acceleration value, in G-force (g), of the accelerometer.
	MinimumAcceleration	Gets the minimum acceleration value, in G-force (g), of the accelerometer.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerCapabilities.AccelerationResolution Property

Note

This property is available only when developing for Zune.

Gets the acceleration resolution, in G-force (g), of the accelerometer.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float AccelerationResolution { get; set; }
```

Property Value

Resolution of the accelerometer.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerCapabilities.HasXAxis Property

Note

This property is available only when developing for Zune.

Indicates if the accelerometer has an X-axis.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasXAxis { get; set; }
```

Property Value

true if measurements along the X-axis of the accelerometer are possible; otherwise **false**.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerCapabilities.HasYAxis Property

Note

This property is available only when developing for Zune.

Indicates if the accelerometer has an Y-axis.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasYAxis { get; set; }
```

Property Value

true if measurements along the Y-axis of the accelerometer are possible; otherwise **false**.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerCapabilities.HasZAxis Property

Note

This property is available only when developing for Zune.

Indicates if the accelerometer has an Z-axis.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasZAxis { get; set; }
```

Property Value

true if measurements along the Z-axis of the accelerometer are possible; otherwise **false**.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)

AccelerometerCapabilities.IsConnected Property

Note

This property is available only when developing for Zune.

Indicates if an accelerometer device is available for use.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; set; }
```

Property Value

true if an accelerometer is available; otherwise **false**.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerCapabilities.MaximumAcceleration Property

Note

This property is available only when developing for Zune.

Gets the maximum acceleration value, in G-force (g), of the accelerometer.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MaximumAcceleration { get; set; }
```

Property Value

Maximum acceleration of the accelerometer.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerCapabilities.MinimumAcceleration Property

Note

This property is available only when developing for Zune.

Gets the minimum acceleration value, in G-force (g), of the accelerometer.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float MinimumAcceleration { get; set; }
```

Property Value

Minimum acceleration of the accelerometer.

See Also

Reference

[AccelerometerCapabilities Structure](#)

[AccelerometerCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerState Structure

Note

This structure is available only when developing for Zune.

Provides information on the current state of the accelerometer device and a helper function for rotational computation.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct AccelerometerState
```

See Also

Reference

[AccelerometerState Members](#)



[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune



AccelerometerState Members

The following tables list the members exposed by the AccelerometerState type.





Public Constructors

	Name	Description
	AccelerometerState	Overloaded. Initializes a new instance of AccelerometerState .
	AccelerometerState	Initializes a new instance of the AccelerometerState structure with the specified acceleration.



Public Properties

	Name	Description
	Acceleration	Current acceleration, in G-force (g), of the accelerometer device.
	IsConnected	Indicates if an accelerometer device is available for use.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetRotation	Gets the current rotation matrix for the device from the current acceleration data.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AccelerometerState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerState Constructor

Initializes a new instance of [AccelerometerState](#).

Overload List

Name	Description
------	-------------

See Also

Reference

[AccelerometerState Structure](#)

[AccelerometerState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerState Constructor

Note

This constructor is available only when developing for Zune.

Initializes a new instance of the [AccelerometerState](#) structure with the specified acceleration.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AccelerometerState (  
    Vector3 acceleration  
)
```

Parameters

acceleration

Acceleration, in G-force (g), of the new object.

See Also

Reference

[AccelerometerState Structure](#)





[AccelerometerState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



PlatformsZune

AccelerometerState Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetRotation	Gets the current rotation matrix for the device from the current acceleration data.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AccelerometerState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerState.GetRotation Method

Note

This method is available only when developing for Zune.

Gets the current rotation matrix for the device from the current acceleration data.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix GetRotation ()
```

Return Value

Matrix representing the current rotation of the accelerometer device.

Remarks

A common usage of this property is to rotate a 3D model to match the tilt of the device in the real world. For instance, the following line of code sets the basic effect's world matrix to the current rotation matrix of the device:

```
basicEffect.World = accState.GetRotation();
```

See Also

Reference

[AccelerometerState Structure](#)



[AccelerometerState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerState Properties

Public Properties

	Name	Description
	Acceleration	Current acceleration, in G-force (g), of the accelerometer device.
	IsConnected	Indicates if an accelerometer device is available for use.

See Also

Reference

[AccelerometerState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

AccelerometerState.Acceleration Property

Note

This property is available only when developing for Zune.

Current acceleration, in G-force (g), of the accelerometer device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 Acceleration { get; }
```

Property Value

The current acceleration, between [MinimumAcceleration](#) and [MaximumAcceleration](#), of the accelerometer.

See Also

Reference

[AccelerometerState Structure](#)

[AccelerometerState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

AccelerometerState.IsConnected Property

Note

This property is available only when developing for Zune.

Indicates if an accelerometer device is available for use.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

true if an accelerometer is available; otherwise **false**.

See Also

Reference

[AccelerometerState Structure](#)

[AccelerometerState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

Buttons Enumeration

Enumerates input device buttons.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum Buttons
```

Members

Member name	Description
A	A button
B	B button
Back	BACK button
BigButton	Big button
DPadDown	Directional pad up
DPadLeft	Directional pad left
DPadRight	Directional pad right
DPadUp	Directional pad down
LeftShoulder	Left bumper (shoulder) button
LeftStick	Left stick button (pressing the left stick)
LeftThumbstickDown	Left stick is towards down
LeftThumbstickLeft	Left stick is towards the left
LeftThumbstickRight	Left stick is towards the right
LeftThumbstickUp	Left stick is towards up
LeftTrigger	Left trigger
RightShoulder	Right bumper (shoulder) button
RightStick	Right stick button (pressing the right stick)
RightThumbstickDown	Right stick is towards down
RightThumbstickLeft	Right stick is towards the left
RightThumbstickRight	Right stick is towards the right
RightThumbstickUp	Right stick is towards up
RightTrigger	Right trigger
Start	START button
X	X button
Y	Y button

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

[GamePadState.IsButtonDown Method](#)

[GamePadState.IsButtonUp Method](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ButtonState Enumeration

Identifies the state of a controller button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum ButtonState
```

Members

Member name	Description
Pressed	The button is pressed.
Released	The button is released.

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePad Class

Allows retrieval of user interaction with an Xbox 360 Controller and setting of controller vibration motors.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class GamePad
```

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePad Members](#)









[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



GamePad Members

The following tables list the members exposed by the GamePad type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCapabilities	Retrieves the capabilities of an Xbox 360 Controller.
	GetHashCode	(Inherited from Object .)
	GetState	Overloaded. Gets the current state of a game pad controller. As an option, it specifies a dead zone processing method for the analog sticks.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetVibration	Sets the vibration motor speeds on an Xbox 360 Controller.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)









Reference

[GamePad Class](#)



[Microsoft.Xna.Framework.Input Namespace](#)

GamePad Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCapabilities	Retrieves the capabilities of an Xbox 360 Controller.
	GetHashCode	(Inherited from Object .)
	GetState	Overloaded. Gets the current state of a game pad controller. As an option, it specifies a dead zone processing method for the analog sticks.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetVibration	Sets the vibration motor speeds on an Xbox 360 Controller.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePad Class](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePad.GetCapabilities Method

Retrieves the capabilities of an Xbox 360 Controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static GamePadCapabilities GetCapabilities (  
    PlayerIndex playerIndex  
)
```

Parameters

playerIndex

Index of the controller to query.

Return Value

The capabilities of the controller.

Exceptions

Exception type	Condition
InvalidOperationException	An unknown error has occurred. Verify that the correct <i>playerIndex</i> is being requested.

See Also

Reference

[GamePad Class](#)

[GamePad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePad.GetState Method

Gets the current state of a game pad controller. As an option, it specifies a dead zone processing method for the analog sticks.

Overload List

Name	Description
GamePad.GetState (PlayerIndex)	Gets the current state of a game pad controller.
GamePad.GetState (PlayerIndex, GamePad DeadZone)	Gets the current state of a game pad controller, using a specified dead zone on an analog stick positions.

Remarks When you call [GetState](#), check the returned structure's **IsConnected** property value. If the controller has been disconnected, this value will be **false**. When a controller is disconnected, values for its state will not be valid. Also, the controller will not receive any new vibration settings.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePad Class](#)

[GamePad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePad.GetState Method (PlayerIndex)

Gets the current state of a game pad controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static GamePadState GetState (
    PlayerIndex playerIndex
)
```

Parameters

playerIndex

Player index for the controller you want to query.

Return Value

The current state of the controller.

Exceptions

Exception type	Condition
InvalidOperationException	An unknown error has occurred. Verify that the correct <i>playerIndex</i> is being requested.

Remarks

★ Best Practice

- On the first menu presented by the game, consider asking the player to sign in by pressing **A** or **Start**. This enables the game to detect the correct controller in use for each player in the game.
- When processing player input, you can examine [SignedInGamer.GameDefaults](#) to determine a player's preferences for how the Xbox 360 controller is used to send input to the game.

When you call [GetState](#), check the returned structure's **IsConnected** property value. If the controller has been disconnected, this value will be **false**. When a controller is disconnected, values for its state will not be valid. Also, the controller will not receive any new vibration settings.

This method uses the [GamePadDeadZone.IndependentAxes](#) dead zone to smooth the values of the analog sticks. To specify a different dead zone type, or to specify no-dead-zone processing, pass a [GamePadDeadZone](#) value to this method.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[SignedInGamer.PlayerIndex Property](#)

[SignedInGamer.GameDefaults](#)

[GamePad Class](#)

[GamePad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePad.GetState Method (PlayerIndex, GamePadDeadZone)

Gets the current state of a game pad controller, using a specified dead zone on analog stick positions.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static GamePadState GetState (
    PlayerIndex playerIndex,
    GamePadDeadZone deadZoneMode
)
```

Parameters

playerIndex

Player index for the controller you want to query.

deadZoneMode

Enumerated value that specifies what dead zone type to use.

Return Value

The current state of the controller.

Exceptions

Exception type	Condition
InvalidOperationException	An unknown error has occurred. Verify that the correct <i>playerIndex</i> is being requested.

Remarks

★ Best Practice

- On the first menu presented by the game, consider asking the player to sign in by pressing **A** or **Start**. This enables the game to detect the correct controller in use for each player in the game.
- When processing player input, you can examine [SignedInGamer.GameDefaults](#) to determine a player's preferences for how the Xbox 360 controller is used to send input to the game.

When you call [GetState](#), check the returned structure's **IsConnected** property value. If the controller has been disconnected, this value will be **false**. When a controller is disconnected, values for its state will not be valid. Also, the controller will not receive any new vibration settings.

The *deadZoneMode* parameter specifies the type of dead-zone processing to apply to the analog sticks on the controller. By applying a dead zone, controller inputs will report values near the center and edges of their ranges more accurately. If you wish to do your own dead-zone processing, use **GamePadDeadZone.None**.

On Zune, the default is "no dead zone." Dead zones are not necessary on Zune. You can specify a dead zone if you want, but you need to remember this setting will reduce the available input in a simple device.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[SignedInGamer.PlayerIndex Property](#)

[SignedInGamer.GameDefaults](#)

[GamePad Class](#)

[GamePad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePad.SetVibration Method

Sets the vibration motor speeds on an Xbox 360 Controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool SetVibration (
    PlayerIndex playerIndex,
    float leftMotor,
    float rightMotor
)
```

Parameters

playerIndex

Player index that identifies the controller to set.

leftMotor

The speed of the left motor, between 0.0 and 1.0. This motor is a low-frequency motor.

rightMotor

The speed of the right motor, between 0.0 and 1.0. This motor is a high-frequency motor.

Return Value

true if the vibration motors were successfully set; **false** if the controller was unable to process the request.

Exceptions

Exception type	Condition
InvalidOperationException	An unknown error has occurred. Verify that the correct <i>playerIndex</i> is being requested.

Remarks

If this method returns **false**, it means that the controller is either disconnected or too busy to accept the new vibration settings. It is not advisable to loop in wait for this method to return **true**.

See Also

Tasks

[Tutorial 2: Making Your Model Move Using Input](#)

[How To: Detect Whether a Controller Button Is Pressed](#)

Reference

[GamePad Class](#)

[GamePad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons Structure

Identifies whether the buttons on an Xbox 360 Controller are pressed or released.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamePadButtons
```

Remarks

★ Best Practice

- If your game allows the player to control acceleration using the Xbox 360 controller, you may check [GameDefaults.AccelerateWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control acceleration.
- If your game allows the player to control braking using the Xbox 360 controller, you may check [GameDefaults.BrakeWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control braking.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune












GamePadButtons Members

The following tables list the members exposed by the GamePadButtons type.








Public Constructors

Name	Description
 GamePadButtons	Initializes a new instance of the GamePadButtons class, setting the specified buttons to pressed.



Public Properties

Name	Description
 A	Identifies whether the A button on the Xbox 360 Controller is pressed.
 B	Identifies whether the B button on the Xbox 360 Controller is pressed.
 Back	Identifies whether the BACK button on the Xbox 360 Controller is pressed.
 BigButton	Identifies whether the BigButton button is pressed.
 LeftShoulder	Identifies whether the left shoulder (bumper) button on the Xbox 360 Controller is pressed.
 LeftStick	Identifies whether the left stick button on the Xbox 360 Controller is pressed (the stick is "clicked" in).
 RightShoulder	Identifies whether the right shoulder (bumper) button on the Xbox 360 Controller is pressed.
 RightStick	Identifies whether the right stick button on the Xbox 360 Controller is pressed (the stick is "clicked" in).
 Start	Identifies whether the START button on the Xbox 360 Controller is pressed.
 X	Identifies whether the X button on the Xbox 360 Controller is pressed.
 Y	Identifies whether the Y button on the Xbox 360 Controller is pressed.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Sop_Equality	Determines whether two GamePadButtons instances are equal.
 Sop_Inequality	Determines whether two GamePadButtons instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePadButtons Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadButtons Constructor

Initializes a new instance of the [GamePadButtons](#) class, setting the specified buttons to pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadButtons (  
    Buttons buttons  
)
```

Parameters

buttons

Buttons to initialize as pressed. Specify a single button, or combine multiple buttons using a bitwise OR operation.

Remarks

Note that [GamePadButtons](#) only stores the state of digital buttons (buttons that are pressed or unpressed, such as **A** or right bumper). Trigger or stick values specified in the *buttons* argument are ignored.

Games normally use [GetState](#) to get the true gamepad state. This constructor is used to instead simulate gamepad input for passing within the game's own input subsystem.

See Also

Reference

[GamePadButtons Structure](#)








[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether two GamePadButtons instances are equal.
	Op_Inequality	Determines whether two GamePadButtons instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadButtons Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadButtons.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GamePadButtons.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GamePadButtons.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadButtons.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.op_Equality Method

Determines whether two [GamePadButtons](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GamePadButtons left,  
    GamePadButtons right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.op_Inequality Method

Determines whether two [GamePadButtons](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GamePadButtons left,  
    GamePadButtons right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GamePadButtons Structure](#)












[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons Properties

Public Properties

	Name	Description
	A	Identifies whether the A button on the Xbox 360 Controller is pressed.
	B	Identifies whether the B button on the Xbox 360 Controller is pressed.
	Back	Identifies whether the BACK button on the Xbox 360 Controller is pressed.
	BigButton	Identifies whether the BigButton button is pressed.
	LeftShoulder	Identifies whether the left shoulder (bumper) button on the Xbox 360 Controller is pressed.
	LeftStick	Identifies whether the left stick button on the Xbox 360 Controller is pressed (the stick is "clicked" in).
	RightShoulder	Identifies whether the right shoulder (bumper) button on the Xbox 360 Controller is pressed.
	RightStick	Identifies whether the right stick button on the Xbox 360 Controller is pressed (the stick is "clicked" in).
	Start	Identifies whether the START button on the Xbox 360 Controller is pressed.
	X	Identifies whether the X button on the Xbox 360 Controller is pressed.
	Y	Identifies whether the Y button on the Xbox 360 Controller is pressed.

See Also

Reference

[GamePadButtons Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadButtons.A Property

Identifies whether the **A** button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState A { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

Remarks

★ Best Practice

The **A** button is usually used to confirm the selection of an item.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.B Property

Identifies whether the **B** button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState B { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

Remarks

★ Best Practice

The **B** button is usually used to cancel or back out of a menu in a game.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.Back Property

Identifies whether the **BACK** button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Back { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

Remarks

★ Best Practice

During gameplay, the **Back** button might be used to bring up a scoreboard or other contextual information such as a map.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.BigButton Property

Identifies whether the **BigButton** button is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState BigButton { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.LeftShoulder Property

Identifies whether the left shoulder (bumper) button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState LeftShoulder { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.LeftStick Property

Identifies whether the left stick button on the Xbox 360 Controller is pressed (the stick is "clicked" in).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState LeftStick { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.RightShoulder Property

Identifies whether the right shoulder (bumper) button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState RightShoulder { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.RightStick Property

Identifies whether the right stick button on the Xbox 360 Controller is pressed (the stick is "clicked" in).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState RightStick { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.Start Property

Identifies whether the **START** button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Start { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

Remarks

★ Best Practice

During gameplay, the **Start** button is usually used to pause the game.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.X Property

Identifies whether the **X** button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState X { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadButtons.Y Property

Identifies whether the **Y** button on the Xbox 360 Controller is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Y { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadButtons Structure](#)

[GamePadButtons Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities Structure

Describes the capabilities of an Xbox 360 Controller, including controller type and whether the controller supports voice.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamePadCapabilities
```

See Also

Reference

[GamePadCapabilities Members](#)



























[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




GamePadCapabilities Members

The following tables list the members exposed by the GamePadCapabilities type.



Public Properties

Name	Description
 GamePadType	Gets the type of controller.
 HasAButton	Indicates whether the controller has an A button.
 HasBackButton	Indicates whether the controller has a BACK button.
 HasBButton	Indicates whether the controller has a B button.
 HasBigButton	Indicates whether the controller has a BigButton button.
 HasDPadDownButton	Indicates whether the controller has a directional pad DOWN button.
 HasDPadLeftButton	Indicates whether the controller has a directional pad LEFT button.
 HasDPadRightButton	Indicates whether the controller has a directional pad RIGHT button.
 HasDPadUpButton	Indicates whether the controller has a directional pad UP button.
 HasLeftShoulderButton	Indicates whether the controller has a left bumper button.
 HasLeftStickButton	Indicates whether the controller has a digital button control on the left analog stick.
 HasLeftTrigger	Indicates whether the controller has a left analog trigger.
 HasLeftVibrationMotor	Indicates whether the controller has a low-frequency vibration motor.
 HasLeftXThumbStick	Indicates whether the controller supports a left analog control with horizontal movement.
 HasLeftYThumbStick	Indicates whether the controller supports a left analog control with vertical movement.
 HasRightShoulderButton	Indicates whether the controller has a right bumper button.
 HasRightStickButton	Indicates whether the controller has a digital button control on the right analog stick.
 HasRightTrigger	Indicates whether the controller has a right analog trigger.
 HasRightVibrationMotor	Indicates whether the controller has a high-frequency vibration motor.
 HasRightXThumbStick	Indicates whether the controller supports a right analog control with horizontal movement.
 HasRightYThumbStick	Indicates whether the controller supports a right analog control with vertical movement.
 HasStartButton	Indicates whether the controller has a START button.
 HasVoiceSupport	Indicates whether the controller supports voice.
 HasXButton	Indicates whether the controller has an X button.
 HasYButton	Indicates whether the controller has a Y button.
 IsConnected	Indicates whether the Xbox 360 Controller is connected.

Public Methods

Name	Description
 Equals	(Inherited from Object.)
 GetType	(Inherited from Object.)
 ReferenceEquals	(Inherited from Object.)

Protected Methods

Name	Description
 Finalize	(Inherited from Object.)
 MemberwiseClone	(Inherited from Object.)

See Also




Reference

[GamePadCapabilities Structure](#)



[Microsoft.Xna.Framework.Input Namespace](#)

GamePadCapabilities Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



























Reference

[GamePadCapabilities Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadCapabilities Properties

Public Properties

	Name	Description
	GamePadType	Gets the type of controller.
	HasAButton	Indicates whether the controller has an A button.
	HasBackButton	Indicates whether the controller has a BACK button.
	HasBButton	Indicates whether the controller has a B button.
	HasBigButton	Indicates whether the controller has a BigButton button.
	HasDPadDownButton	Indicates whether the controller has a directional pad DOWN button.
	HasDPadLeftButton	Indicates whether the controller has a directional pad LEFT button.
	HasDPadRightButton	Indicates whether the controller has a directional pad RIGHT button.
	HasDPadUpButton	Indicates whether the controller has a directional pad UP button.
	HasLeftShoulderButton	Indicates whether the controller has a left bumper button.
	HasLeftStickButton	Indicates whether the controller has a digital button control on the left analog stick.
	HasLeftTrigger	Indicates whether the controller has a left analog trigger.
	HasLeftVibrationMotor	Indicates whether the controller has a low-frequency vibration motor.
	HasLeftXThumbStick	Indicates whether the controller supports a left analog control with horizontal movement.
	HasLeftYThumbStick	Indicates whether the controller supports a left analog control with vertical movement.
	HasRightShoulderButton	Indicates whether the controller has a right bumper button.
	HasRightStickButton	Indicates whether the controller has a digital button control on the right analog stick.
	HasRightTrigger	Indicates whether the controller has a right analog trigger.
	HasRightVibrationMotor	Indicates whether the controller has a high-frequency vibration motor.
	HasRightXThumbStick	Indicates whether the controller supports a right analog control with horizontal movement.
	HasRightYThumbStick	Indicates whether the controller supports a right analog control with vertical movement.
	HasStartButton	Indicates whether the controller has a START button.
	HasVoiceSupport	Indicates whether the controller supports voice.
	HasXButton	Indicates whether the controller has an X button.
	HasYButton	Indicates whether the controller has a Y button.
	IsConnected	Indicates whether the Xbox 360 Controller is connected.

See Also

Reference

[GamePadCapabilities Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadCapabilities.GamePadType Property

Gets the type of controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadType GamePadType { get; }
```

Property Value

Enumerated value describing the controller type.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasAButton Property

Indicates whether the controller has an **A** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasAButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasBackButton Property

Indicates whether the controller has a **BACK** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasBackButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasBButton Property

Indicates whether the controller has a **B** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasBButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasBigButton Property

Indicates whether the controller has a **BigButton** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasBigButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasDPadDownButton Property

Indicates whether the controller has a directional pad DOWN button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasDPadDownButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasDPadLeftButton Property

Indicates whether the controller has a directional pad LEFT button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasDPadLeftButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasDPadRightButton Property

Indicates whether the controller has a directional pad RIGHT button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasDPadRightButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasDPadUpButton Property

Indicates whether the controller has a directional pad UP button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasDPadUpButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasLeftShoulderButton Property

Indicates whether the controller has a left bumper button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftShoulderButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasLeftStickButton Property

Indicates whether the controller has a digital button control on the left analog stick.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftStickButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasLeftTrigger Property

Indicates whether the controller has a left analog trigger.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftTrigger { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasLeftVibrationMotor Property

Indicates whether the controller has a low-frequency vibration motor.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftVibrationMotor { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasLeftXThumbStick Property

Indicates whether the controller supports a left analog control with horizontal movement.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftXThumbStick { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasLeftYThumbStick Property

Indicates whether the controller supports a left analog control with vertical movement.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftYThumbStick { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasRightShoulderButton Property

Indicates whether the controller has a right bumper button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasRightShoulderButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasRightStickButton Property

Indicates whether the controller has a digital button control on the right analog stick.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasRightStickButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasRightTrigger Property

Indicates whether the controller has a right analog trigger.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasRightTrigger { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasRightVibrationMotor Property

Indicates whether the controller has a high-frequency vibration motor.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasRightVibrationMotor { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasRightXThumbStick Property

Indicates whether the controller supports a right analog control with horizontal movement.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasRightXThumbStick { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasRightYThumbStick Property

Indicates whether the controller supports a right analog control with vertical movement.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasRightYThumbStick { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasStartButton Property

Indicates whether the controller has a **START** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasStartButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasVoiceSupport Property

Indicates whether the controller supports voice.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasVoiceSupport { get; }
```

Property Value

true if the controller supports voice; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasXButton Property

Indicates whether the controller has an **X** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasXButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.HasYButton Property

Indicates whether the controller has a **Y** button.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasYButton { get; }
```

Property Value

true if the controller has this capability; **false** otherwise.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadCapabilities.IsConnected Property

Indicates whether the Xbox 360 Controller is connected.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

true if the controller is connected; **false** otherwise.

Remarks When calling [GetState](#) or [GetCapabilities](#), check the returned structure's **IsConnected** property value. If the controller has been disconnected, this value will be **false**. When a controller is disconnected, values for its state and/or capabilities will not be valid. The controller will also not be able to receive new vibration settings.

See Also

Reference

[GamePadCapabilities Structure](#)

[GamePadCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDeadZone Enumeration

Specifies a type of dead zone processing to apply to Xbox 360 controller analog sticks when calling [GetState](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GamePadDeadZone
```

Members

Member name	Description
Circular	The combined X and Y position of each stick is compared against the dead zone. This provides better control than IndependentAxes when the stick is used as a two-dimensional control surface, such as when controlling a character's view in a first-person game.
IndependentAxes	The X and Y positions of each stick are compared against the dead zone independently. This setting is the default when calling GetState .
None	The values of each stick are not processed and are returned by GetState as "raw" values. This is best if you intend to implement your own dead zone processing.

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad Structure

Identifies which directions on the directional pad of an Xbox 360 Controller are being pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamePadDPad
```

Remarks

★ Best Practice

Many games allow movement to be controlled with either the left thumbstick or the directional pad (d-pad). The left thumbstick may be accessed with [ThumbSticks](#), and the directional pad is available through [DPad](#).

See Also

Reference

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune





GamePadDPad Members

The following tables list the members exposed by the GamePadDPad type.








Public Constructors

Name	Description
 GamePadDPad	Initializes a new instance of the GamePadDPad class.



Public Properties

Name	Description
 Down	Identifies whether the Down direction on the Xbox 360 Controller directional pad is pressed.
 Left	Identifies whether the Left direction on the Xbox 360 Controller directional pad is pressed.
 Right	Identifies whether the Right direction on the Xbox 360 Controller directional pad is pressed.
 Up	Identifies whether the Up direction on the Xbox 360 Controller directional pad is pressed.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two GamePadDPad instances are equal.
 Op_Inequality	Determines whether two GamePadDPad instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadDPad Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadDPad Constructor

Initializes a new instance of the [GamePadDPad](#) class.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadDPad (  
    ButtonState upValue,  
    ButtonState downValue,  
    ButtonState leftValue,  
    ButtonState rightValue  
)
```

Parameters

upValue

Directional pad up button state.

downValue

Directional pad down button state.

leftValue

Directional pad left button state.

rightValue

Directional pad right button state.

Remarks

Games normally use [GetState](#) to get the true gamepad state. This constructor is used to instead simulate gamepad input for passing within the game's own input subsystem.

See Also

Reference

[GamePadDPad Structure](#)










[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether two GamePadDPad instances are equal.
	 op_Inequality	Determines whether two GamePadDPad instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadDPad Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadDPad.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GamePadDPad.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GamePadDPad.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadDPad.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.op_Equality Method

Determines whether two [GamePadDPad](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GamePadDPad left,  
    GamePadDPad right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.op_Inequality Method

Determines whether two [GamePadDPad](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GamePadDPad left,  
    GamePadDPad right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GamePadDPad Structure](#)





[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad Properties

Public Properties

	Name	Description
	Down	Identifies whether the Down direction on the Xbox 360 Controller directional pad is pressed.
	Left	Identifies whether the Left direction on the Xbox 360 Controller directional pad is pressed.
	Right	Identifies whether the Right direction on the Xbox 360 Controller directional pad is pressed.
	Up	Identifies whether the Up direction on the Xbox 360 Controller directional pad is pressed.

See Also

Reference

[GamePadDPad Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadDPad.Down Property

Identifies whether the Down direction on the Xbox 360 Controller directional pad is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Down { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.Left Property

Identifies whether the Left direction on the Xbox 360 Controller directional pad is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Left { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.Right Property

Identifies whether the Right direction on the Xbox 360 Controller directional pad is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Right { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadDPad.Up Property

Identifies whether the Up direction on the Xbox 360 Controller directional pad is pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState Up { get; }
```

Property Value

[ButtonState.Pressed](#) if the button is pressed; [ButtonState.Released](#) otherwise.

See Also

Reference

[GamePadDPad Structure](#)

[GamePadDPad Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState Structure

Represents specific information about the state of an Xbox 360 Controller, including the current state of buttons and sticks.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamePadState
```

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePadState Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







GamePadState Members

The following tables list the members exposed by the GamePadState type.










Public Constructors

Name	Description
 GamePadState	Overloaded. Initializes a new instance of the GamePadState class.



Public Properties

Name	Description
 Buttons	Returns a structure that identifies what buttons on the Xbox 360 controller are pressed.
 DPad	Returns a structure that identifies what directions of the directional pad on the Xbox 360 Controller are pressed.
 IsConnected	Indicates whether the Xbox 360 Controller is connected.
 PacketNumber	Gets the packet number associated with this state.
 ThumbSticks	Returns a structure that indicates the position of the Xbox 360 Controller sticks (thumbsticks).
 Triggers	Returns a structure that identifies the position of triggers on the Xbox 360 controller.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 IsButtonDown	Determines whether specified input device buttons are pressed in this GamePadState .
 IsButtonUp	Determines whether specified input device buttons are up (not pressed) in this GamePadState .
 IsOp_Equality	Determines whether two GamePadState instances are equal.
 IsOp_Inequality	Determines whether two GamePadState instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePadState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadState Constructor

Initializes a new instance of the [GamePadState](#) class.

Overload List

Name	Description
GamePadState (GamePadThumbSticks, GamePadTriggers, GamePadButtons, GamePadDPad)	Initializes a new instance of the GamePadState class using the specified GamePadThumbSticks , GamePadTriggers , GamePadButtons , and GamePadDPad .
GamePadState (Vector2, Vector2, Single, Single, Buttons[])	Initializes a new instance of the GamePadState class with the specified stick, trigger, and button values.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadState Constructor (GamePadThumbSticks, GamePadTriggers, GamePadButtons, GamePadDPad)

Initializes a new instance of the [GamePadState](#) class using the specified [GamePadThumbSticks](#), [GamePadTriggers](#), [GamePadButtons](#), and [GamePadDPad](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadState (  
    GamePadThumbSticks thumbSticks,  
    GamePadTriggers triggers,  
    GamePadButtons buttons,  
    GamePadDPad dPad  
)
```

Parameters

thumbSticks

Initial thumbstick state.

triggers

Initial trigger state.

buttons

Initial button state.

dPad

Initial directional pad state.

Remarks

Games normally use [GetState](#) to get the true gamepad state. This constructor is used to instead simulate gamepad input for passing within the game's own input subsystem.

Thumbstick state is different from the state indicated by the [GamePadButtons.LeftStick Property](#) or the [GamePadButtons.RightStick Property](#). Thumbstick state refers to the position of the thumbstick, while the **LeftStick** or **RightStick** properties indicate whether the stick is "clicked" in.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState Constructor (Vector2, Vector2, Single, Single, Buttons[])

Initializes a new instance of the [GamePadState](#) class with the specified stick, trigger, and button values.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadState (  
    Vector2 leftThumbStick,  
    Vector2 rightThumbStick,  
    float leftTrigger,  
    float rightTrigger,  
    Buttons[] buttons  
)
```

Parameters

leftThumbStick

Left stick value. Each axis is clamped between -1.0 and 1.0.

rightThumbStick

Right stick value. Each axis is clamped between -1.0 and 1.0.

leftTrigger

Left trigger value. This value is clamped between 0.0 and 1.0.

rightTrigger

Right trigger value. This value is clamped between 0.0 and 1.0.

buttons

Array or parameter list of [Buttons](#) to initialize as pressed.

Remarks

Games normally use [GetState](#) to get the true gamepad state. This constructor is used to instead simulate gamepad input for passing within the game's own input subsystem.

See Also

Reference

[GamePadState Structure](#)












[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	IsButtonDown	Determines whether specified input device buttons are pressed in this GamePadState .
	IsButtonUp	Determines whether specified input device buttons are up (not pressed) in this GamePadState .
 	op_Equality	Determines whether two GamePadState instances are equal.
 	op_Inequality	Determines whether two GamePadState instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadState.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GamePadState.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GamePadState.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadState.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.IsButtonDown Method

Determines whether specified input device buttons are pressed in this [GamePadState](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsButtonDown (  
    Buttons button  
)
```

Parameters

button

Buttons to query. Specify a single button, or combine multiple buttons using a bitwise OR operation.

Return Value

true if all specified buttons are pressed; **false** otherwise.

Remarks

For digital buttons (such as **A**, **START**, and right bumper) the return value indicates whether the button is pressed.

For analog inputs (triggers and thumbsticks), the return value indicates whether the input exceeds dead zone values. For example, a trigger will not be reported as pressed until it is pulled about 12% of its full range. **IsButtonDown** always uses the default dead zone processing mode, [GamePadDeadZone.IndependentAxes](#).

If multiple buttons are specified (by combining them using the bitwise OR operator), then *all* must be pressed (or exceed dead zone values) in order for **IsButtonDown** to return **true**.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[GamePadState.IsButtonDown Method](#)

[GamePad.GetState Method](#)

[GamePadDeadZone](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.IsButtonUp Method

Determines whether specified input device buttons are up (not pressed) in this [GamePadState](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsButtonUp (  
    Buttons button  
)
```

Parameters

button

Buttons to query. Specify a single button, or combine multiple buttons using a bitwise OR operation.

Return Value

true if any specified buttons are up; **false** otherwise.

Remarks

For digital buttons (such as **A**, **START**, and right bumper) the return value indicates whether the button is up (not pressed).

For analog inputs (triggers and thumbsticks), the return value indicates whether the input exceeds dead zone values. For example, a trigger will be reported as not pressed until it is pulled about 12% of its full range. **IsButtonUp** always uses the default dead zone processing mode, [GamePadDeadZone.IndependentAxes](#).

If multiple buttons are specified (by combining them using the bitwise OR operator), **IsButtonUp** returns **true** if *any* of the specified buttons are not pressed (or do not exceed dead zone values).

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[GamePadState.IsButtonDown Method](#)

[GamePad.GetState Method](#)

[GamePadDeadZone](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.op_Equality Method

Determines whether two [GamePadState](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GamePadState left,  
    GamePadState right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.op_Inequality Method

Determines whether two [GamePadState](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GamePadState left,  
    GamePadState right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GamePadState Structure](#)







[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState Properties

Public Properties

	Name	Description
	Buttons	Returns a structure that identifies what buttons on the Xbox 360 controller are pressed.
	DPad	Returns a structure that identifies what directions of the directional pad on the Xbox 360 Controller are pressed.
	IsConnected	Indicates whether the Xbox 360 Controller is connected.
	PacketNumber	Gets the packet number associated with this state.
	ThumbSticks	Returns a structure that indicates the position of the Xbox 360 Controller sticks (thumbsticks).
	Triggers	Returns a structure that identifies the position of triggers on the Xbox 360 controller.

See Also

Reference

[GamePadState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadState.Buttons Property

Returns a structure that identifies what buttons on the Xbox 360 controller are pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadButtons Buttons { get; }
```

Property Value

A structure that identifies buttons pressed on the controller.

Remarks

★ Best Practice

- If your game allows the player to control acceleration using the Xbox 360 controller, you may check [GameDefaults.AccelerateWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control acceleration.
- If your game allows the player to control braking using the Xbox 360 controller, you may check [GameDefaults.BrakeWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control braking.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GameDefaults.AccelerateWithButtons Property](#)

[GameDefaults.BrakeWithButtons Property](#)

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.DPad Property

Returns a structure that identifies what directions of the directional pad on the Xbox 360 Controller are pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadDPad DPad { get; }
```

Property Value

Structure that identifies what directional pad directions are pressed.

Remarks

★ Best Practice

Many games allow movement to be controlled with either the left thumbstick or the directional pad (d-pad). The left thumbstick may be accessed with [GamePadState.ThumbSticks](#).

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.IsConnected Property

Indicates whether the Xbox 360 Controller is connected.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

true if the controller is connected; **false** otherwise.

Remarks When calling [GetState](#) or [GetCapabilities](#), check the returned structure's **IsConnected** property value. If the controller has been disconnected, this value will be **false**. When a controller is disconnected, values for its state and/or capabilities will not be valid. The controller will also not be able to receive new vibration settings.

See Also

Tasks

[How To: Pause a Game](#)

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.PacketNumber Property

Gets the packet number associated with this state.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PacketNumber { get; }
```

Property Value

The packet number associated with this state.

Remarks You can use [PacketNumber](#) to determine whether input state has changed. If the value of [PacketNumber](#) remains the same between two sequential calls to [GetState](#), then there has been no change in input.

See Also

Tasks

[How To: Detect Whether a Controller Button Is Pressed](#)

[How To: Detect Whether a Controller Is Disconnected](#)

Concepts

[Input Overview](#)

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.ThumbSticks Property

Returns a structure that indicates the position of the Xbox 360 Controller sticks (thumbsticks).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadThumbSticks ThumbSticks { get; }
```

Property Value

Structure that indicates the position of the sticks.

Remarks

★ Best Practice

You can check [GameDefaults.MoveWithRightThumbStick](#) to determine if the player prefers using the right or left thumbstick to control movement.

Many games allow movement to be controlled with either the left thumbstick or the directional pad (d-pad). The directional pad is available through [GamePadState.DPad](#).

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadState.Triggers Property

Returns a structure that identifies the position of triggers on the Xbox 360 controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadTriggers Triggers { get; }
```

Property Value

Structure that identifies the position of the triggers.

Remarks

★ Best Practice

- If your game allows the player to control acceleration using the Xbox 360 controller, you may check [GameDefaults.AccelerateWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control acceleration.
- If your game allows the player to control braking using the Xbox 360 controller, you may check [GameDefaults.BrakeWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control braking.

See Also

Reference

[GamePadState Structure](#)

[GamePadState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks Structure

Structure that represents the position of left and right sticks (thumbsticks) on an Xbox 360 Controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamePadThumbSticks
```

Remarks

★ Best Practice

You can check [GameDefaults.MoveWithRightThumbStick](#) to determine if the player prefers using the right or left thumbstick to control movement.

See Also

Reference

[GamePadThumbSticks Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



GamePadThumbSticks Members

The following tables list the members exposed by the GamePadThumbSticks type.








Public Constructors

Name	Description
 GamePadThumbSticks	Initializes a new instance of the GamePadThumbSticks class.



Public Properties

Name	Description
 Left	Returns the position of the left Xbox 360 Controller stick (thumbstick) as a 2D vector.
 Right	Returns the position of the right Xbox 360 Controller stick (thumbstick) as a 2D vector.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 op_Equality	Determines whether two GamePadThumbSticks instances are equal.
 op_Inequality	Determines whether two GamePadThumbSticks instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadThumbSticks Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadThumbSticks Constructor

Initializes a new instance of the [GamePadThumbSticks](#) class.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadThumbSticks (  
    Vector2 leftThumbstick,  
    Vector2 rightThumbstick  
)
```

Parameters

leftThumbstick

Left stick value. Each axis is clamped between -1.0 and 1.0.

rightThumbstick

Right stick value. Each axis is clamped between -1.0 and 1.0.

Remarks

Games normally use [GetState](#) to get the true gamepad state. This constructor is used to instead simulate gamepad input for passing within the game's own input subsystem.

See Also

Reference

[GamePadThumbSticks Structure](#)










[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether two GamePadThumbSticks instances are equal.
	 op_Inequality	Determines whether two GamePadThumbSticks instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadThumbSticks Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadThumbSticks.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GamePadThumbSticks.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GamePadThumbSticks.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadThumbSticks.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks.op_Equality Method

Determines whether two [GamePadThumbSticks](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GamePadThumbSticks left,  
    GamePadThumbSticks right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks.op_Inequality Method

Determines whether two [GamePadThumbSticks](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GamePadThumbSticks left,  
    GamePadThumbSticks right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GamePadThumbSticks Structure](#)



[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks Properties

Public Properties

	Name	Description
	Left	Returns the position of the left Xbox 360 Controller stick (thumbstick) as a 2D vector.
	Right	Returns the position of the right Xbox 360 Controller stick (thumbstick) as a 2D vector.

See Also

Reference

[GamePadThumbSticks Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadThumbSticks.Left Property

Returns the position of the left Xbox 360 Controller stick (thumbstick) as a 2D vector.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 Left { get; }
```

Property Value

Two-dimensional vector that identifies the position of the left stick. Each axis is represented as a floating-point value from -1.0 to 1.0 .

Remarks

★Best Practice

You can check [GameDefaults.MoveWithRightThumbStick](#) to determine if the player prefers using the right or left thumbstick to control movement.

Each of the thumbstick axis members is a signed value describing the position of the thumbstick. A value of 0 is centered. Negative values signify down or to the left. Positive values signify up or to the right. The controller typically does not generate thumbstick values within a dead zone, or marginal input region around the center position.

Example

```
if (GamePad.GetState( PlayerIndex.One ).ThumbSticks.Left.Y == 1.0f)
{
    // Player one has pressed the left thumbstick up.
}
```

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadThumbSticks.Right Property

Returns the position of the right Xbox 360 Controller stick (thumbstick) as a 2D vector.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 Right { get; }
```

Property Value

Two-dimensional vector that identifies the position of the right stick. Each axis is represented as a floating-point value from -1.0 to 1.0.

Remarks

★ Best Practice

You can check [GameDefaults.MoveWithRightThumbStick](#) to determine if the player prefers using the right or left thumbstick to control movement.

See Also

Reference

[GamePadThumbSticks Structure](#)

[GamePadThumbSticks Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers Structure

Structure that defines the position of the left and right triggers on an Xbox 360 controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct GamePadTriggers
```

Remarks

★ Best Practice

- If your game allows the player to control acceleration using the Xbox 360 controller, you may check [GameDefaults.AccelerateWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control acceleration.
- If your game allows the player to control braking using the Xbox 360 controller, you may check [GameDefaults.BrakeWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control braking.

See Also

Reference

[GamePadTriggers Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



GamePadTriggers Members

The following tables list the members exposed by the GamePadTriggers type.








Public Constructors

	Name	Description
	GamePadTriggers	Initializes a new instance of the GamePadTriggers class.



Public Properties

	Name	Description
	Left	Identifies the position of the left trigger on the Xbox 360 Controller.
	Right	Identifies the position of the right trigger on the Xbox 360 Controller.

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GamePadTriggers instances are equal.
	op_Inequality	Determines whether two GamePadTriggers instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadTriggers Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadTriggers Constructor

Initializes a new instance of the [GamePadTriggers](#) class.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamePadTriggers (  
    float leftTrigger,  
    float rightTrigger  
)
```

Parameters

leftTrigger

Left trigger value. This value is clamped between 0.0 and 1.0.

rightTrigger

Right trigger value. This value is clamped between 0.0 and 1.0.

Remarks

Games normally use [GetState](#) to get the true gamepad state. This constructor is used to instead simulate gamepad input for passing within the game's own input subsystem.

See Also

Reference

[GamePadTriggers Structure](#)








[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two GamePadTriggers instances are equal.
	op_Inequality	Determines whether two GamePadTriggers instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamePadTriggers Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadTriggers.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
GamePadTriggers.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
GamePadTriggers.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadTriggers.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers.op_Equality Method

Determines whether two [GamePadTriggers](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    GamePadTriggers left,  
    GamePadTriggers right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers.op_Inequality Method

Determines whether two [GamePadTriggers](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    GamePadTriggers left,  
    GamePadTriggers right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[GamePadTriggers Structure](#)



[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers Properties

Public Properties

	Name	Description
	Left	Identifies the position of the left trigger on the Xbox 360 Controller.
	Right	Identifies the position of the right trigger on the Xbox 360 Controller.

See Also

Reference

[GamePadTriggers Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

GamePadTriggers.Left Property

Identifies the position of the left trigger on the Xbox 360 Controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Left { get; }
```

Property Value

Position of the left trigger, as a value between 0.0 and 1.0.

Remarks

★ Best Practice

- If your game allows the player to control acceleration using the Xbox 360 controller, you may check [GameDefaults.AccelerateWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control acceleration.
- If your game allows the player to control braking using the Xbox 360 controller, you may check [GameDefaults.BrakeWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control braking.

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadTriggers.Right Property

Identifies the position of the right trigger on the Xbox 360 Controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Right { get; }
```

Property Value

Position of the right trigger, as a value between 0.0 and 1.0.

Remarks

★ Best Practice

- If your game allows the player to control acceleration using the Xbox 360 controller, you may check [GameDefaults.AccelerateWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control acceleration.
- If your game allows the player to control braking using the Xbox 360 controller, you may check [GameDefaults.BrakeWithButtons](#) to determine if the player prefers using the gamepad buttons or triggers to control braking.

See Also

Reference

[GamePadTriggers Structure](#)

[GamePadTriggers Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamePadType Enumeration

Describes the type of a specified Xbox 360 controller.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum GamePadType
```

Members

Member name	Description
AlternateGuitar	Controller is an alternate guitar
ArcadeStick	Controller is an arcade stick
BigButtonPad	Controller is a big button pad
DancePad	Controller is a dance pad
DrumKit	Controller is a drum kit
FlightStick	Controller is a flight stick
GamePad	Controller is the Xbox 360 Controller
Guitar	Controller is a guitar
Unknown	Controller is an unknown type
Wheel	Controller is a wheel

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Keyboard Class

Allows retrieval of keystrokes from a keyboard input device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class Keyboard
```

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[Keyboard Members](#)








[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



Keyboard Members

The following tables list the members exposed by the Keyboard type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	 GetState	Overloaded. Returns the current keyboard or Chatpad state.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)








Reference

[Keyboard Class](#)



[Microsoft.Xna.Framework.Input Namespace](#)

Keyboard Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
 	GetState	Overloaded. Returns the current keyboard or Chatpad state.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Keyboard Class](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Keyboard.GetState Method

Returns the current keyboard or Chatpad state.

Overload List

Name	Description
Keyboard.GetState ()	Returns the current keyboard state.
Keyboard.GetState (PlayerIndex)	Returns the current Chatpad state for the specified player.

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[Keyboard Class](#)

[Keyboard Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Keyboard.GetState Method ()

Returns the current keyboard state.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static KeyboardState GetState ()
```

Return Value

Current keyboard state.

Exceptions

Exception type	Condition
InvalidOperationException	An invalid operation occurred when querying the keyboard state.

Remarks

This method queries an attached USB keyboard. To query a Chatpad instead, use [Keyboard.GetState \(PlayerIndex\)](#).

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[Keyboard Class](#)

[Keyboard Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[Keyboard.GetState Method](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Keyboard.GetState Method (PlayerIndex)

Returns the current Chatpad state for the specified player.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static KeyboardState GetState (  
    PlayerIndex playerIndex  
)
```

Parameters

playerIndex

Player index of the Chatpad to query.

Return Value

Current Chatpad state.

Exceptions

Exception type	Condition
InvalidOperationException	An invalid operation occurred when querying the Chatpad state.

Remarks

Each player can have his or her own Chatpad, which is queried as a keyboard. To query an attached USB keyboard instead, use the [GetState](#) method that does not require a player index.

This method works on Xbox 360 platforms only. Windows does not include driver support for the ChatPad.

This method is included in the assemblies for Windows, but it is effectively a null function on that platform. Calls to `Keyboard.GetState(playerIndex)` from a Windows application are accepted, but will never return a ChatPad state change (key press or release).

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[Keyboard Class](#)

[Keyboard Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState Structure

Represents a state of keystrokes recorded by a keyboard input device.

Zune Specific Information

Keyboard state information is not supported, and any values returned do not reflect the current keyboard state.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct KeyboardState
```

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


KeyboardState Members

The following tables list the members exposed by the KeyboardState type.










Public Constructors

Name	Description
 KeyboardState	Initializes a new instance of the KeyboardState class.



Public Properties

Name	Description
 Item	Returns the state of a particular key.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetPressedKeys	Gets an array of values that correspond to the keyboard keys that are currently being pressed.
 GetType	(Inherited from Object .)
 IsKeyDown	Returns whether a specified key is currently being pressed.
 IsKeyUp	Returns whether a specified key is currently not pressed.
 op_Equality	Compares two objects to determine whether they are the same.
 op_Inequality	Compares two objects to determine whether they are different.
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[KeyboardState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

KeyboardState Constructor

Initializes a new instance of the [KeyboardState](#) class.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public KeyboardState (  
    Keys[] keys  
)
```

Parameters

keys

Array or parameter list of [Keys](#) to initialize as pressed.

Remarks

Games normally use [GetState](#) to get the true keyboard state. This constructor is used to instead simulate keyboard input for passing within the game's own input subsystem.

See Also

Reference

[KeyboardState Structure](#)












[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetPressedKeys	Gets an array of values that correspond to the keyboard keys that are currently being pressed.
	GetType	(Inherited from Object .)
	IsKeyDown	Returns whether a specified key is currently being pressed.
	IsKeyUp	Returns whether a specified key is currently not pressed.
	 op_Equality	Compares two objects to determine whether they are the same.
	 op_Inequality	Compares two objects to determine whether they are different.
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[KeyboardState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

KeyboardState.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
KeyboardState.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
KeyboardState.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

KeyboardState.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object to compare this object to.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState.GetPressedKeys Method

Gets an array of values that correspond to the keyboard keys that are currently being pressed.

Zune Specific Information

Keyboard state information is not supported. This method always returns an empty array.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Keys[] GetPressedKeys ()
```

Return Value

The keys that are currently pressed.

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState.IsKeyDown Method

Returns whether a specified key is currently being pressed.

Zune Specific Information

Keyboard state information is not supported. This method always returns **false**.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsKeyDown (  
    Keys key  
)
```

Parameters

key

Enumerated value that specifies the key to query.

Return Value

true if the key specified by *key* is being held down; **false** otherwise.

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState.IsKeyUp Method

Returns whether a specified key is currently not pressed.

Zune Specific Information

Keyboard state information is not supported. This method always returns **true**.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsKeyUp (  
    Keys key  
)
```

Parameters

key

Enumerated value that specifies the key to query.

Return Value

true if the key specified by *key* is not pressed; **false** otherwise.

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState.op_Equality Method

Compares two objects to determine whether they are the same.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    KeyboardState a,  
    KeyboardState b  
)
```

Parameters

a

Object to the left of the equality operator.

b

Object to the right of the equality operator.

Return Value

true if the objects are the same; **false** otherwise.

See Also

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState.op_Inequality Method

Compares two objects to determine whether they are different.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    KeyboardState a,  
    KeyboardState b  
)
```

Parameters

a

Object to the left of the inequality operator.

b

Object to the right of the inequality operator.

Return Value

true if the objects are different; **false** otherwise.

See Also

Reference

[KeyboardState Structure](#)


[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyboardState Properties

Public Properties

	Name	Description
	Item	Returns the state of a particular key.

See Also

Reference

[KeyboardState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

KeyboardState.Item Property

Returns the state of a particular key.

Zune Specific Information

Keyboard state information is not supported. This method always returns [KeyState.Up](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public KeyState this [
    Keys key
] { get; }
```

Property Value

The state of the key specified by *key*.

See Also

Tasks

[How To: Detect Whether a Key Is Pressed](#)

Concepts

[Input Overview](#)

Reference

[KeyboardState Structure](#)

[KeyboardState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Keys Enumeration

Identifies a particular key on a keyboard.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum Keys
```

Members

Member name	Description
A	A key
Add	Add key
Apps	Applications key
Attn	Attn key
B	B key
Back	BACKSPACE key
BrowserBack	Windows 2000/XP: Browser Back key
BrowserFavorites	Windows 2000/XP: Browser Favorites key
BrowserForward	Windows 2000/XP: Browser Forward key
BrowserHome	Windows 2000/XP: Browser Start and Home key
BrowserRefresh	Windows 2000/XP: Browser Refresh key
BrowserSearch	Windows 2000/XP: Browser Search key
BrowserStop	Windows 2000/XP: Browser Stop key
C	C key
CapsLock	CAPS LOCK key
ChatPadGreen	Green ChatPad key
ChatPadOrange	Orange ChatPad key
CrSel	CrSel key
D	D key
D0	Used for miscellaneous characters; it can vary by keyboard.
D1	Used for miscellaneous characters; it can vary by keyboard.
D2	Used for miscellaneous characters; it can vary by keyboard.
D3	Used for miscellaneous characters; it can vary by keyboard.
D4	Used for miscellaneous characters; it can vary by keyboard.
D5	Used for miscellaneous characters; it can vary by keyboard.
D6	Used for miscellaneous characters; it can vary by keyboard.
D7	Used for miscellaneous characters; it can vary by keyboard.
D8	Used for miscellaneous characters; it can vary by keyboard.
D9	Used for miscellaneous characters; it can vary by keyboard.
Decimal	Decimal key
Delete	DEL key
Divide	Divide key
Down	DOWN ARROW key
E	E key
End	END key
Enter	ENTER key
EraseEof	Erase EOF key
Escape	ESC key
Execute	EXECUTE key
Exsel	ExSel key
F	F key

F1	F1 key
F10	F10 key
F11	F11 key
F12	F12 key
F13	F13 key
F14	F14 key
F15	F15 key
F16	F16 key
F17	F17 key
F18	F18 key
F19	F19 key
F2	F2 key
F20	F20 key
F21	F21 key
F22	F22 key
F23	F23 key
F24	F24 key
F3	F3 key
F4	F4 key
F5	F5 key
F6	F6 key
F7	F7 key
F8	F8 key
F9	F9 key
G	G key
H	H key
Help	HELP key
Home	HOME key
I	I key
ImeConvert	IME Convert key
ImeNoConvert	IME NoConvert key
Insert	INS key
J	J key
K	K key
Kana	Kana key on Japanese keyboards
Kanji	Kanji key on Japanese keyboards
L	L key
LaunchApplication1	Windows 2000/XP: Start Application 1 key
LaunchApplication2	Windows 2000/XP: Start Application 2 key
LaunchMail	Windows 2000/XP: Start Mail key
Left	LEFT ARROW key
LeftAlt	Left ALT key
LeftControl	Left CONTROL key
LeftShift	Left SHIFT key
LeftWindows	Left Windows key
M	M key
MediaNextTrack	Windows 2000/XP: Next Track key
MediaPlayPause	Windows 2000/XP: Play/Pause Media key
MediaPreviousTrack	Windows 2000/XP: Previous Track key
MediaStop	Windows 2000/XP: Stop Media key
Multiply	Multiply key
N	N key
None	Reserved
NumLock	NUM LOCK key

NumPad0	Numeric keypad 0 key
NumPad1	Numeric keypad 1 key
NumPad2	Numeric keypad 2 key
NumPad3	Numeric keypad 3 key
NumPad4	Numeric keypad 4 key
NumPad5	Numeric keypad 5 key
NumPad6	Numeric keypad 6 key
NumPad7	Numeric keypad 7 key
NumPad8	Numeric keypad 8 key
NumPad9	Numeric keypad 9 key
O	O key
Oem8	Used for miscellaneous characters; it can vary by keyboard.
OemAuto	OEM Auto key
OemBackslash	Windows 2000/XP: The OEM angle bracket or backslash key on the RT 102 key keyboard
OemClear	CLEAR key
OemCloseBrackets	Windows 2000/XP: The OEM close bracket key on a US standard keyboard
OemComma	Windows 2000/XP: For any country/region, the ',' key
OemCopy	OEM Copy key
OemEnlW	OEM Enlarge Window key
OemMinus	Windows 2000/XP: For any country/region, the '-' key
OemOpenBrackets	Windows 2000/XP: The OEM open bracket key on a US standard keyboard
OemPeriod	Windows 2000/XP: For any country/region, the '.' key
OemPipe	Windows 2000/XP: The OEM pipe key on a US standard keyboard
OemPlus	Windows 2000/XP: For any country/region, the '+' key
OemQuestion	Windows 2000/XP: The OEM question mark key on a US standard keyboard
OemQuotes	Windows 2000/XP: The OEM singled/double quote key on a US standard keyboard
OemSemicolon	Windows 2000/XP: The OEM Semicolon key on a US standard keyboard
OemTilde	Windows 2000/XP: The OEM tilde key on a US standard keyboard
P	P key
Pa1	PA1 key
PageDown	PAGE DOWN key
PageUp	PAGE UP key
Pause	PAUSE key
Play	Play key
Print	PRINT key
PrintScreen	PRINT SCREEN key
ProcessKey	Windows 95/98/Me, Windows NT 4.0, Windows 2000/XP: IME PROCESS key
Q	Q key
R	R key
Right	RIGHT ARROW key
RightAlt	Right ALT key
RightControl	Right CONTROL key
RightShift	Right SHIFT key
RightWindows	Right Windows key
S	S key
Scroll	SCROLL LOCK key
Select	SELECT key
SelectMedia	Windows 2000/XP: Select Media key
Separator	Separator key
Sleep	Computer Sleep key
Space	SPACEBAR
Subtract	Subtract key
T	T key
Tab	TAB key

U	U key
Up	UP ARROW key
V	V key
VolumeDown	Windows 2000/XP: Volume Down key
VolumeMute	Windows 2000/XP: Volume Mute key
VolumeUp	Windows 2000/XP: Volume Up key
W	W key
X	X key
Y	Y key
Z	Z key
Zoom	Zoom key

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

KeyState Enumeration

Identifies the state of a keyboard key.

Zune Specific Information

Keyboard state information is not supported. This value of this enumeration is always **KeyState.Up**.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum KeyState
```

Members

Member name	Description
Down	The key is pressed.
Up	The key is released.

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Mouse Class

Allows retrieval of position and button clicks from a mouse input device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class Mouse
```

Xbox 360 Specific Information

Mouse functionality is not supported, but mouse-related code will compile and execute. Values returned by these calls are not accurate.

Zune Specific Information

Mouse functionality is not supported but mouse-related code will compile and execute. Values returned by these calls are not accurate.

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[Mouse Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune










Mouse Members

The following tables list the members exposed by the Mouse type.



Public Properties

	Name	Description
	WindowHandle	Gets or sets the window used for mouse processing. Mouse coordinates returned by GetState are relative to the upper-left corner of this window.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
 	GetState	Gets the current state of the mouse, including mouse position and buttons pressed.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
 	SetPosition	Sets the position of the mouse cursor relative to the upper-left corner of the window.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)










Reference

[Mouse Class](#)





[Microsoft.Xna.Framework.Input Namespace](#)

Mouse Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
 	GetState	Gets the current state of the mouse, including mouse position and buttons pressed.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
 	SetPosition	Sets the position of the mouse cursor relative to the upper-left corner of the window.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
 	Finalize	(Inherited from Object .)
 	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Mouse Class](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Mouse.GetState Method

Gets the current state of the mouse, including mouse position and buttons pressed.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static MouseState GetState ()
```

Return Value

Current state of the mouse.

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[Mouse Class](#)

[Mouse Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Mouse.SetPosition Method

Sets the position of the mouse cursor relative to the upper-left corner of the window.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void SetPosition (  
    int x,  
    int y  
)
```

Parameters

x
The horizontal position of the mouse cursor, relative to the left edge of the game window.

y
The vertical position of the mouse cursor, relative to the upper edge of the game window.

Remarks

When using this method to take relative input, such as in a first-person game, set the position of the mouse to the center of your game window each frame. This will allow you to read mouse movement on both axes with the greatest amount of granularity.

See Also

Reference

[Mouse Class](#)


[Mouse Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Mouse Properties

Public Properties

	Name	Description
	WindowHandle	Gets or sets the window used for mouse processing. Mouse coordinates returned by GetState are relative to the upper-left corner of this window.

See Also

Reference

[Mouse Class](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Mouse.WindowHandle Property

Gets or sets the window used for mouse processing. Mouse coordinates returned by [GetState](#) are relative to the upper-left corner of this window.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IntPtr WindowHandle { get; set; }
```

Property Value

Handle to a window.

Remarks

By default, this property is set to the game window ([Game.Window Property](#)). To select a different window as the coordinate reference, set **WindowHandle** to that window.

If the mouse is above or to the left of the reference window, the returned coordinates are negative. If the mouse is below or to the right of the reference window, the returned coordinates are greater than the window's client area.

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[Mouse Class](#)

[Mouse Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState Structure

Represents the state of a mouse input device, including mouse cursor position and buttons pressed.

Xbox 360 Specific Information

The state of any mouse button is always [ButtonState.Released](#), and the current mouse position and scrollwheel value are always 0.

Xbox 360 Specific Information

The state of any mouse button is always [ButtonState.Released](#) and the current mouse position and scrollwheel values are always 0.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]
public struct MouseState
```

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









MouseState Members

The following tables list the members exposed by the MouseState type.








Public Constructors

Name	Description
 MouseState	Initializes a new instance of the MouseState class.



Public Properties

Name	Description
 LeftButton	Returns the state of the left mouse button.
 MiddleButton	Returns the state of the middle mouse button.
 RightButton	Returns the state of the right mouse button.
 ScrollWheelValue	Gets the cumulative mouse scroll wheel value since the game was started.
 X	Specifies the horizontal position of the mouse cursor.
 XButton1	Returns the state of XBUTTON1.
 XButton2	Returns the state of XBUTTON2.
 Y	Specifies the vertical position of the mouse cursor.

Public Methods

Name	Description
 Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two MouseState instances are equal.
 Op_Inequality	Determines whether two MouseState instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Retrieves a string representation of this object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[MouseState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

MouseState Constructor

Initializes a new instance of the [MouseState](#) class.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MouseState (  
    int x,  
    int y,  
    int scrollWheel,  
    ButtonState leftButton,  
    ButtonState middleButton,  
    ButtonState rightButton,  
    ButtonState xButton1,  
    ButtonState xButton2  
)
```

Parameters

x

Horizontal mouse position.

y

Vertical mouse position.

scrollWheel

Mouse scroll wheel value.

leftButton

Left mouse button state.

middleButton

Middle mouse button state.

rightButton

Right mouse button state.

xButton1

XBUTTON1 state.

xButton2

XBUTTON2 state.

Remarks

Games normally use [GetState](#) to get the true mouse state. This constructor is used to instead simulate mouse input for passing within the game's own input subsystem.

See Also

Reference

[MouseState Structure](#)








[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState Methods

Public Methods

	Name	Description
	Equals	Overloaded. Returns a value that indicates whether the current instance is equal to a specified object.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether two MouseState instances are equal.
	op_Inequality	Determines whether two MouseState instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Retrieves a string representation of this object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MouseState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

MouseState.Equals Method

Returns a value that indicates whether the current instance is equal to a specified object.

Overload List

Name	Description
MouseState.Equals (Object)	Returns a value that indicates whether the current instance is equal to a specified object.
MouseState.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

MouseState.Equals Method (Object)

Returns a value that indicates whether the current instance is equal to a specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; **false** otherwise.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.op_Equality Method

Determines whether two [MouseState](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    MouseState left,  
    MouseState right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the instances are equal; **false** otherwise.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.op_Inequality Method

Determines whether two [MouseState](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    MouseState left,  
    MouseState right  
)
```

Parameters

left

Object on the left of the equal sign.

right

Object on the right of the equal sign.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.ToString Method

Retrieves a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[MouseState Structure](#)









[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState Properties

Public Properties

	Name	Description
	LeftButton	Returns the state of the left mouse button.
	MiddleButton	Returns the state of the middle mouse button.
	RightButton	Returns the state of the right mouse button.
	ScrollWheelValue	Gets the cumulative mouse scroll wheel value since the game was started.
	X	Specifies the horizontal position of the mouse cursor.
	XButton1	Returns the state of XBUTTON1.
	XButton2	Returns the state of XBUTTON2.
	Y	Specifies the vertical position of the mouse cursor.

See Also

Reference

[MouseState Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

MouseState.LeftButton Property

Returns the state of the left mouse button.

Xbox 360 Specific Information

The value of this property is always [ButtonState.Released](#).

Zune Specific Information

The value of this property is always [ButtonState.Released](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState LeftButton { get; }
```

Property Value

State of the left mouse button.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.MiddleButton Property

Returns the state of the middle mouse button.

Xbox 360 Specific Information

The value of this property is always [ButtonState.Released](#).

Zune Specific Information

The value of this property is always [ButtonState.Released](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState MiddleButton { get; }
```

Property Value

State of the middle mouse button.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.RightButton Property

Returns the state of the right mouse button.

Xbox 360 Specific Information

The value of this property is always [ButtonState.Released](#).

Zune Specific Information

The value of this property is always [ButtonState.Released](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState RightButton { get; }
```

Property Value

State of the right mouse button.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.ScrollWheelValue Property

Gets the cumulative mouse scroll wheel value since the game was started.

Xbox 360 Specific Information

The value of this property is always 0.

Zune Specific Information

The value of this property is always 0.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ScrollWheelValue { get; }
```

Property Value

The scroll wheel value.

Remarks

Unlike standard Windows and WinForms mouse messages, which report a relative value for the scroll wheel, XNA reports the total scroll wheel value accumulated over the length of a game's execution. To process a wheel movement, you must compare the previous frame's wheel value to the current wheel value.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.X Property

Specifies the horizontal position of the mouse cursor.

Xbox 360 Specific Information

The value of this property is always 0.

Zune Specific Information

The value of this property is always 0.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int X { get; }
```

Property Value

Horizontal position of the mouse cursor in relation to the upper-left corner of the game window.

Remarks

If the mouse is to the left of the client area (the inside portion of the game window, excluding the borders), the returned coordinate is negative. If the mouse is to the right of the client area, the returned coordinate is greater than the window's client area width.

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.XButton1 Property

Returns the state of XBUTTON1.

Xbox 360 Specific Information

The value of this property is always [ButtonState.Released](#).

Zune Specific Information

The value of this property is always [ButtonState.Released](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState XButton1 { get; }
```

Property Value

The state of XBUTTON1.

Remarks XBUTTON1 and XBUTTON2 are additional buttons used on many mouse devices, often for forward and backward navigation in Web browsers. They return the same data as standard mouse buttons.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.XButton2 Property

Returns the state of XBUTTON2.

Xbox 360 Specific Information

The value of this property is always [ButtonState.Released](#).

Zune Specific Information

The value of this property is always [ButtonState.Released](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ButtonState XButton2 { get; }
```

Property Value

The state of XBUTTON2.

Remarks XBUTTON1 and XBUTTON2 are additional buttons used on many mouse devices, often for forward and backward navigation in Web browsers. They return the same data as standard mouse buttons.

See Also

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MouseState.Y Property

Specifies the vertical position of the mouse cursor.

Xbox 360 Specific Information

The value of this property is always 0.

Zune Specific Information

The value of this property is always 0.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Y { get; }
```

Property Value

Vertical position of the mouse cursor in relation to the upper-left corner of the game window.

Remarks

If the mouse is above the client area (the inside portion of the game window, excluding the borders), the returned coordinate is negative. If the mouse is below the client area, the returned coordinate is greater than the window's client area height.

See Also

Tasks

[How To: Get the Current Mouse Position \(Windows\)](#)

Concepts

[Input Overview](#)

Reference

[MouseState Structure](#)

[MouseState Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

TouchCollection Structure

Note

This structure is available only when developing for Zune.

Provides methods and properties for accessing state information for the touch screen of a Zune device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct TouchCollection : IList<TouchLocation>, ICollection<TouchLocation>, IEnumerable<TouchLocation>, IEnumerableable
```

Remarks

This structure contains information on the current state of all touch locations detected by the touch device. This information consists of:

- Positions of up to 8 touch locations.
- Amount of pressure for each touch location.
- Previous position, if any, of the current touch locations.

See Also

Reference

[TouchCollection Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)





TouchCollection Members

The following tables list the members exposed by the TouchCollection type.














Public Constructors

Name	Description
 TouchCollection	Initializes a new instance of the TouchCollection structure with a set of touch locations.



Public Properties

Name	Description
 Count	Gets the current number of touch locations for the touch screen.
 IsConnected	Indicates if the touch screen is available for use.
 IsReadOnly	Determines if the touch location array is read-only.
 Item	Gets or sets the information of the specified touch location.


Public Methods

Name	Description
 Add	Adds a TouchLocation to the collection.
 Clear	Removes all TouchLocation objects from the collection.
 Contains	Checks whether the current touch collection contains the specified touch location.
 CopyTo	Copies the touch location to the collection, at the specified index.
 Equals	(Inherited from Object .)
 FindById	Retrieves the touch location whose ID matches the specified ID.
 GetEnumerator	Returns an enumerator that iterates through the TouchCollection .
 GetType	(Inherited from Object .)
 IndexOf	Determines the index of a TouchLocation in the TouchCollection .
 Insert	Inserts a new TouchLocation into the TouchCollection .
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the specified TouchLocation from the TouchCollection .
 RemoveAt	Removes the TouchLocation item found at the specified location from the TouchCollection .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the TouchCollection .

See Also

Reference

[TouchCollection Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchCollection Constructor

Note

This constructor is available only when developing for Zune.

Initializes a new instance of the TouchCollection structure with a set of touch locations.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TouchCollection (  
    TouchLocation[] touches  
)
```

Parameters

touches

Array of touch locations.

See Also

Reference

[TouchCollection Structure](#)














[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



[PlatformsZune](#)

TouchCollection Methods


Public Methods

Name	Description
 Add	Adds a TouchLocation to the collection.
 Clear	Removes all TouchLocation objects from the collection.
 Contains	Checks whether the current touch collection contains the specified touch location.
 CopyTo	Copies the touch location to the collection, at the specified index.
 Equals	(Inherited from Object .)
 FindById	Retrieves the touch location whose ID matches the specified ID.
 GetEnumerator	Returns an enumerator that iterates through the TouchCollection .
 GetType	(Inherited from Object .)
 IndexOf	Determines the index of a TouchLocation in the TouchCollection .
 Insert	Inserts a new TouchLocation into the TouchCollection .
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the specified TouchLocation from the TouchCollection .
 RemoveAt	Removes the TouchLocation item found at the specified location from the TouchCollection .

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the TouchCollection .

See Also

Reference

[TouchCollection Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchCollection.Add Method

Note

This method is available only when developing for Zune.

Adds a [TouchLocation](#) to the collection.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Add (  
    TouchLocation item  
)
```

Parameters

item

The [TouchLocation](#) to add.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)

TouchCollection.Clear Method

Note

This method is available only when developing for Zune.

Removes all [TouchLocation](#) objects from the collection.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Clear ()
```

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Contains Method

Note

This method is available only when developing for Zune.

Checks whether the current touch collection contains the specified touch location.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Contains (  
    TouchLocation item  
)
```

Parameters

item

The touch location to be checked against the current collection.

Return Value

true if the item is present in the collection; otherwise **false**.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.CopyTo Method

Note

This method is available only when developing for Zune.

Copies the touch location to the collection, at the specified index.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void CopyTo (  
    TouchLocation[] array,  
    int arrayIndex  
)
```

Parameters

array

Array receiving the copied touch location.

arrayIndex

Target index of the collection.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.FindById Method

Note

This method is available only when developing for Zune.

Retrieves the touch location whose ID matches the specified ID.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool FindById (
    int id,
    out TouchLocation touchLocation
)
```

Parameters

id

ID of touch location sought.

touchLocation

[[OutAttribute](#)] Touch location item whose ID matches the specified ID.

Return Value

true if the item was found; otherwise **false**.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)

TouchCollection.GetEnumerator Method

Note

This method is available only when developing for Zune.

Returns an enumerator that iterates through the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<TouchLocation> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the touch collection.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.IndexOf Method

Note

This method is available only when developing for Zune.

Determines the index of a [TouchLocation](#) in the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int IndexOf (  
    TouchLocation item  
)
```

Parameters

item

[TouchLocation](#) to be located in the collection.

Return Value

The index of the specified [TouchLocation](#) if found; otherwise –1.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Insert Method

Note

This method is available only when developing for Zune.

Inserts a new [TouchLocation](#) into the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Insert (  
    int index,  
    TouchLocation item  
)
```

Parameters

index

Index in the touch collection for the new item.

item

Item to be inserted.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Remove Method

Note

This method is available only when developing for Zune.

Removes the specified [TouchLocation](#) from the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Remove (  
    TouchLocation item  
)
```

Parameters

item

Item to be removed.

Return Value

true if the item was successfully removed; otherwise **false**.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.RemoveAt Method

Note

This method is available only when developing for Zune.

Removes the [TouchLocation](#) item found at the specified location from the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void RemoveAt (  
    int index  
)
```

Parameters

index

Zero-based index of the item to be removed.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

System.Collections.IEnumerable.GetEnumerator Method

Note

This method is available only when developing for Zune.

Returns an enumerator that iterates through the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) for the touch collection.

See Also

Reference

[TouchCollection Structure](#)





[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection Properties

Public Properties

	Name	Description
	Count	Gets the current number of touch locations for the touch screen.
	IsConnected	Indicates if the touch screen is available for use.
	IsReadOnly	Determines if the touch location array is read-only.
	Item	Gets or sets the information of the specified touch location.

See Also

Reference

[TouchCollection Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchCollection.Count Property

Note

This property is available only when developing for Zune.

Gets the current number of touch locations for the touch screen.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of current locations. The maximum number of trackable locations is 8.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.IsConnected Property

Note

This property is available only when developing for Zune.

Indicates if the touch screen is available for use.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

true if the touch screen is available; otherwise **false**.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)

TouchCollection.IsReadOnly Property

Note

This property is available only when developing for Zune.

Determines if the touch location array is read-only.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

true if the array is read-only; otherwise **false**.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Item Property

Note

This property is available only when developing for Zune.

Gets or sets the information of the specified touch location.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TouchLocation this [
    int index
] { get; set; }
```

Property Value

Zero-based index of a touch location.

See Also

Reference

[TouchCollection Structure](#)

[TouchCollection Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)

TouchCollection.Enumerator Structure

Note

This structure is available only when developing for Zune.

Provides the ability to iterate through the touch locations in a [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct TouchCollection.Enumerator : IEnumerable<TouchLocation>, IDisposable, IEnumera  
tor
```

See Also

Reference

[TouchCollection.Enumerator Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune






TouchCollection.Enumerator Members

The following tables list the members exposed by the TouchCollection.Enumerator type.



Public Properties

	Name	Description
	Current	Gets the current element in the TouchCollection .



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the TouchCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the TouchCollection as an object.
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the TouchCollection .

See Also






Reference

[TouchCollection.Enumerator Structure](#)



[Microsoft.Xna.Framework.Input Namespace](#)

TouchCollection.Enumerator Methods



Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveNext	Advances the enumerator to the next element of the TouchCollection .
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerator.Current	Gets the current element in the TouchCollection as an object.
	System.Collections.IEnumerator.Reset	Sets the enumerator to its initial position, which is before the first element in the TouchCollection .

See Also

Reference

[TouchCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchCollection.Enumerator.Dispose Method

Note

This method is available only when developing for Zune.

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[TouchCollection.Enumerator Structure](#)

[TouchCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Enumerator.MoveNext Method

Note

This method is available only when developing for Zune.

Advances the enumerator to the next element of the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool MoveNext ()
```

Return Value

true if the enumerator was successfully advanced to the next element; **false** if the enumerator passed the end of the collection.

See Also

Reference

[TouchCollection.Enumerator Structure](#)

[TouchCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

System.Collections.IEnumerator.Reset Method

Note

This method is available only when developing for Zune.

Sets the enumerator to its initial position, which is before the first element in the [TouchCollection](#) .

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.IEnumerator.Reset ()
```

See Also

Reference

[TouchCollection.Enumerator Structure](#)


[TouchCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Enumerator Properties

Public Properties

	Name	Description
	Current	Gets the current element in the TouchCollection .

See Also

Reference

[TouchCollection.Enumerator Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchCollection.Enumerator.Current Property

Note

This property is available only when developing for Zune.

Gets the current element in the [TouchCollection](#).

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TouchLocation Current { get; }
```

Property Value

The current element in the collection.

See Also

Reference

[TouchCollection.Enumerator Structure](#)

[TouchCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchCollection.Enumerator.System.Collections.IEnumerator.Current Property

Note

This property is available only when developing for Zune.

Gets the current element in the [TouchCollection](#) as an object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private Object System.Collections.IEnumerator.Current { get; }
```

Property Value

The current element in the collection as an [Object](#).

See Also

Reference

[TouchCollection.Enumerator Structure](#)

[TouchCollection.Enumerator Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation Structure

Note

This structure is available only when developing for Zune.

Provides methods and properties for interacting with a touch location on a touch screen device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct TouchLocation : IEquatable<TouchLocation>
```

Remarks

The **TouchLocation** structure contains information, such as screen location, current state, and amount of pressure, for a registered touch location item. A collection of current touch locations is stored in the [TouchCollection](#) object of the touch screen device.

See Also

Reference

[TouchLocation Members](#)


[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune





TouchLocation Members

The following tables list the members exposed by the TouchLocation type.









Public Constructors

Name	Description
 TouchLocation	Overloaded. Initializes a new instance of the TouchLocation structure.



Public Properties

Name	Description
 Id	Gets the ID of the touch location object.
 Position	Gets the position of the touch location object.
 Pressure	Gets the recorded pressure, in G-force (g), of the touch location.
 State	Gets the state of the touch location.

Public Methods

Name	Description
 Equals	Overloaded. Determines whether the current instance is equal to a specified object.
 GetHashCode	gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether two TouchLocation instances are equal.
 Op_Inequality	Determines whether two TouchLocation instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Gets a string representation of this object.
 TryGetPreviousLocation	Attempts to get the previous location for the touch location object.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TouchLocation Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchLocation Constructor

Initializes a new instance of the **TouchLocation** structure.

Overload List

Name	Description
TouchLocation (Int32, TouchLocationState, Vector2, Single)	Initializes a new instance of the TouchLocation structure with the specified values.
TouchLocation (Int32, TouchLocationState, Vector2, Single, TouchLocationState, Vector2, Single)	Initializes a new instance of the TouchLocation structure with the specified values.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchLocation Constructor (Int32, TouchLocationState, Vector2, Single)

Note

This constructor is available only when developing for Zune.

Initializes a new instance of the **TouchLocation** structure with the specified values.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TouchLocation (  
    int id,  
    TouchLocationState state,  
    Vector2 position,  
    float pressure  
)
```

Parameters

id

ID of the new touch location.

state

State of the new touch location.

position

Position, in screen coordinates, of the new touch location.

pressure

Pressure value, measured in G-force (g), of the new touch location.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation Constructor (Int32, TouchLocationState, Vector2, Single, TouchLocationState, Vector2, Single)

Note

This constructor is available only when developing for Zune.

Initializes a new instance of the **TouchLocation** structure with the specified values.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TouchLocation (  
    int id,  
    TouchLocationState state,  
    Vector2 position,  
    float pressure,  
    TouchLocationState previousState,  
    Vector2 previousPosition,  
    float previousPressure  
)
```

Parameters

id

ID of the new touch location.

state

State of the new touch location.

position

Position, in screen coordinates, of the new touch location.

pressure

Pressure value, measured in G-force (g), of the new touch location.

previousState

Previous state for the new touch location.

previousPosition

Previous position for the new touch location.

previousPressure

Previous pressure, in g, for the new touch location.

See Also

Reference

[TouchLocation Structure](#)











[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)



PlatformsZune

TouchLocation Methods

Public Methods

	Name	Description
	Equals	Overloaded. Determines whether the current instance is equal to a specified object.
	GetHashCode	gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether two TouchLocation instances are equal.
	 op_Inequality	Determines whether two TouchLocation instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Gets a string representation of this object.
	TryGetPreviousLocation	Attempts to get the previous location for the touch location object.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TouchLocation Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchLocation.Equals Method

Determines whether the current instance is equal to a specified object.

Overload List

Name	Description
TouchLocation.Equals (Object)	Determines whether the current instance is equal to the specified object.
TouchLocation.Equals (TouchLocation)	Determines whether the current instance is equal to the specified touch location.
TouchLocation.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchLocation.Equals Method (Object)

Note

This method is available only when developing for Zune.

Determines whether the current instance is equal to the specified object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

Object with which to make the comparison.

Return Value

true if the current instance is equal to the specified object; otherwise **false**.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.Equals Method (TouchLocation)

Note

This method is available only when developing for Zune.

Determines whether the current instance is equal to the specified touch location.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    TouchLocation other  
)
```

Parameters

other

Touch location with which to make the comparison.

Return Value

true if the current instance is equal to the specified touch location; otherwise **false**.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.GetHashCode Method

Note

This method is available only when developing for Zune.

gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.op_Equality Method

Note

This method is available only when developing for Zune.

Determines whether two [TouchLocation](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    TouchLocation value1,  
    TouchLocation value2  
)
```

Parameters

value1

[TouchLocation](#) on the left side of the equal sign.

value2

[TouchLocation](#) on the right side of the equal sign.

Return Value

true if the [TouchLocation](#) instances are equal; otherwise **false**.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.op_Inequality Method

Note

This method is available only when developing for Zune.

Determines whether two [TouchLocation](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    TouchLocation value1,  
    TouchLocation value2  
)
```

Parameters

value1

[TouchLocation](#) on the left side of the equal sign.

value2

[TouchLocation](#) on the right side of the equal sign.

Return Value

true if the [TouchLocation](#) instances are equal; otherwise **false**.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)

TouchLocation.ToString Method

Note

This method is available only when developing for Zune.

Gets a string representation of this object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

String representation of this object.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.TryGetPreviousLocation Method

Note

This method is available only when developing for Zune.

Attempts to get the previous location for the touch location object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool TryGetPreviousLocation (  
    out TouchLocation previousLocation  
)
```

Parameters

previousLocation

[[OutAttribute](#)] Previous location data.

Return Value

true if a previous location was retrieved; otherwise **false**.

See Also

Reference

[TouchLocation Structure](#)





[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation Properties

Public Properties

	Name	Description
	Id	Gets the ID of the touch location object.
	Position	Gets the position of the touch location object.
	Pressure	Gets the recorded pressure, in G-force (g), of the touch location.
	State	Gets the state of the touch location.

See Also

Reference

[TouchLocation Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchLocation.Id Property

Note

This property is available only when developing for Zune.

Gets the ID of the touch location object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Id { get; }
```

Property Value

ID of the touch location.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.Position Property

Note

This property is available only when developing for Zune.

Gets the position of the touch location object.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 Position { get; }
```

Property Value

Position, in screen coordinates, of the touch location.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.Pressure Property

Note

This property is available only when developing for Zune.

Gets the recorded pressure, in G-force (g), of the touch location.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Pressure { get; }
```

Property Value

Pressure of the touch location.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocation.State Property

Note

This property is available only when developing for Zune.

Gets the state of the touch location.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TouchLocationState State { get; }
```

Property Value

State of the touch location.

See Also

Reference

[TouchLocation Structure](#)

[TouchLocation Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchLocationState Enumeration

Note

This enumeration is available only when developing for Zune.

Defines the possible states for a touch location.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum TouchLocationState
```

Members

Member name	Description
Invalid	This touch location position is invalid. Typically, you will encounter this state when a new touch location attempts to get the previous state of itself.
Moved	This touch location position was updated or pressed at same position.
Pressed	This touch location position is new.
Released	This touch location position was released.

See Also

Reference

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchPanel Class

Note

This class is available only when developing for Zune.

Provides methods for retrieving touch panel device information.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class TouchPanel
```

See Also

Reference

[TouchPanel Members](#)










[Microsoft.Xna.Framework.Input Namespace](#)

[PlatformsZune](#)



TouchPanel Members

The following tables list the members exposed by the TouchPanel type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
 	GetCapabilities	Gets the touch panel capabilities for an available device.
	GetHashCode	(Inherited from Object .)
 	GetState	Gets the current state of the touch panel.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also










Reference

[TouchPanel Class](#)



[Microsoft.Xna.Framework.Input Namespace](#)

TouchPanel Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
 	GetCapabilities	Gets the touch panel capabilities for an available device.
	GetHashCode	(Inherited from Object .)
 	GetState	Gets the current state of the touch panel.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TouchPanel Class](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchPanel.GetCapabilities Method

Note

This method is available only when developing for Zune.

Gets the touch panel capabilities for an available device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TouchPanelCapabilities GetCapabilities ()
```

Return Value

Information on the capabilities of an available touch panel device.

See Also

Reference

[TouchPanel Class](#)

[TouchPanel Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchPanel.GetState Method

Note

This method is available only when developing for Zune.

Gets the current state of the touch panel.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TouchCollection GetState ()
```

Return Value

State of the touch panel and any existing touch locations.

See Also

Reference

[TouchPanel Class](#)

[TouchPanel Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchPanelCapabilities Structure

Note

This structure is available only when developing for Zune.

Provides access to information about the touch pad device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public struct TouchPanelCapabilities
```

See Also

Reference

[TouchPanelCapabilities Members](#)




[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune




TouchPanelCapabilities Members

The following tables list the members exposed by the TouchPanelCapabilities type.



Public Properties

	Name	Description
	HasPressure	Determines if the touch panel is able to track pressure information along with touch position.
	IsConnected	Indicates if the touch panel device is available for use.
	MaximumTouchCount	Gets or sets the maximum number of touch locations concurrently tracked by the touch pad device.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also




Reference

[TouchPanelCapabilities Structure](#)



[Microsoft.Xna.Framework.Input Namespace](#)

TouchPanelCapabilities Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also




Reference

[TouchPanelCapabilities Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchPanelCapabilities Properties

Public Properties

	Name	Description
	HasPressure	Determines if the touch panel is able to track pressure information along with touch position.
	IsConnected	Indicates if the touch panel device is available for use.
	MaximumTouchCount	Gets or sets the maximum number of touch locations concurrently tracked by the touch pad device.

See Also

Reference

[TouchPanelCapabilities Structure](#)

[Microsoft.Xna.Framework.Input Namespace](#)

TouchPanelCapabilities.HasPressure Property

Note

This property is available only when developing for Zune.

Determines if the touch panel is able to track pressure information along with touch position.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasPressure { get; set; }
```

Property Value

true if the touch panel tracks pressure; otherwise **false**.

See Also

Reference

[TouchPanelCapabilities Structure](#)

[TouchPanelCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchPanelCapabilities.IsConnected Property

Note

This property is available only when developing for Zune.

Indicates if the touch panel device is available for use.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; set; }
```

Property Value

true if a touch panel is available; otherwise **false**.

See Also

Reference

[TouchPanelCapabilities Structure](#)

[TouchPanelCapabilities Members](#)

[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune

TouchPanelCapabilities.MaximumTouchCount Property

Note

This property is available only when developing for Zune.

Gets or sets the maximum number of touch locations concurrently tracked by the touch pad device.

Namespace: Microsoft.Xna.Framework.Input

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaximumTouchCount { get; set; }
```

Property Value

Maximum number of touch locations tracked by the touch panel.

See Also

Reference

[TouchPanelCapabilities Structure](#)

[TouchPanelCapabilities Members](#)






















[Microsoft.Xna.Framework.Input Namespace](#)

PlatformsZune




Microsoft.Xna.Framework.Media Namespace

Contains classes to enumerate, play, and view songs, albums, playlists, and pictures.

Classes

Name	Description
 Album	Provides access to an album in the media library.
 AlbumCollection	A collection of albums in the media library.
 Artist	Provides access to artist information in the media library.
 ArtistCollection	The collection of all artists in the media library.
 Genre	Provides access to genre information in the media library.
 GenreCollection	The collection of all genres in the media library.
 MediaLibrary	Provides access to songs, playlists, and pictures in the device's media library.
 MediaPlayer	Provides methods and properties to play, pause, resume, and stop songs. MediaPlayer also exposes shuffle, repeat, volume, play position, and visualization capabilities.
 MediaQueue	Provides methods and properties to access and control the queue of playing songs.
 MediaSource	Provides methods and properties to access the source or sources from which the media will be read.
 Picture	Provides access to a picture in the media library.
 PictureAlbum	Provides access to a picture album in the media library.
 PictureAlbumCollection	A collection of picture albums in the media library.
 PictureCollection	A collection of pictures in the media library.
 Playlist	Provides access to a playlist in the media library.
 PlaylistCollection	A collection of playlists in the media library.
 Song	Provides access to a song in the song library.
 SongCollection	A collection of songs in the song library.
 Video	Represents a video.
 VideoPlayer	Provides methods and properties to playback, pause, resume, and stop video. VideoPlayer also exposes repeat, volume, and play position information.
 VisualizationData	Encapsulates visualization (frequency and sample) data for the currently-playing song.

Enumerations

Name	Description
 MediaSourceType	Type of the media source.
 MediaState	Media playback state (playing, paused, or stopped).
 VideoSoundtrackType	Type of sounds in a video

Album Class

Provides access to an album in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Album : IEquatable<Album>
```

Remarks

The [Album](#) class provides information about an album, including the album's [Name](#), [Artist](#), and [Songs](#).

You can obtain an **Album** object through the [AlbumCollection.Item](#) indexer and the [Song.Album](#) property.

See Also

Reference

[AlbumCollection.Item](#) Property

[Song.Album](#) Property

[Album Members](#)







[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune











Album Members

The following tables list the members exposed by the Album type.



Public Properties

	Name	Description
	Artist	Gets the Artist of the Album.
	Duration	Gets the duration of the Album.
	Genre	Gets the Genre of the Album.
	HasArt	Gets a value indicating whether the Album has associated album art.
	Name	Gets the name of the Album.
	Songs	Gets a SongCollection that contains the songs on the album.

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Album are equal.
	GetAlbumArt	Gets the album art as a texture.
	GetHashCode	Gets the hash code for this instance.
	GetThumbnail	Gets the album art thumbnail as a texture.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether the specified Album instances are equal.
	Op_Inequality	Determines whether the specified Album instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of this Album.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Album is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also













Reference

[Album Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Album Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Album are equal.
	GetAlbumArt	Gets the album art as a texture.
	GetHashCode	Gets the hash code for this instance.
	GetThumbnail	Gets the album art thumbnail as a texture.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether the specified Album instances are equal.
	 op_Inequality	Determines whether the specified Album instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of this Album.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Album is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Album Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Album.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Equals Method

Determines whether two instances of [Album](#) are equal.

Overload List

Name	Description
Album.Equals (Album)	Determines whether the specified Album is equal to this Album .
Album.Equals (Object)	Determines whether the specified Object is equal to this Album .
Album.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Album.Equals Method (Album)

Determines whether the specified [Album](#) is equal to this **Album**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Album other  
)
```

Parameters

other

Album to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [Album](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.GetAlbumArt Method

Gets the album art as a texture.

Windows Specific Information

Album information is not supported. This property is always set to **null**.

Xbox 360 Specific Information

Album information is not supported. This property is always set to **null**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetAlbumArt (
    IServiceProvider serviceProvider
)
```

Parameters

serviceProvider

Content manager service provider, obtained through the [Game.Services](#) property.

Return Value

Texture with the album art, or **null** if the album has no associated art.

See Also

Reference

[Album.HasArt Property](#)

[Album.GetThumbnail Method](#)

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.GetThumbnail Method

Gets the album art thumbnail as a texture.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetThumbnail (  
    IServiceProvider serviceProvider  
)
```

Parameters

serviceProvider

Content manager service provider, obtained through the [Game.Services](#) property.

Return Value

Texture with the album art thumbnail, or **null** if the album has no associated thumbnail.

See Also

Reference

[Album.HasArt Property](#)

[Album.GetAlbumArt Method](#)

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.op_Equality Method

Determines whether the specified [Album](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Album first,  
    Album second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.op_Inequality Method

Determines whether the specified [Album](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Album first,  
    Album second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.ToString Method

Returns a [String](#) representation of this [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[Album Class](#)







[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album Properties

Public Properties

	Name	Description
	Artist	Gets the Artist of the Album.
	Duration	Gets the duration of the Album.
	Genre	Gets the Genre of the Album.
	HasArt	Gets a value indicating whether the Album has associated album art.
	Name	Gets the name of the Album.
	Songs	Gets a SongCollection that contains the songs on the album.

See Also

Reference

[Album Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Album.Artist Property

Gets the [Artist](#) of the [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Artist Artist { get; }
```

Property Value

[Artist](#) of this [Album](#).

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Duration Property

Gets the duration of the [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

Duration of the album, as a [TimeSpan](#) structure.

Remarks

The duration is the total duration of all of the album's songs that are present on the device. This may be less than the duration attributed to the entire album.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Genre Property

Gets the [Genre](#) of the [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Genre Genre { get; }
```

Property Value

[Genre](#) of this [Album](#).

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.HasArt Property

Gets a value indicating whether the [Album](#) has associated album art.

Windows Specific Information

Album information is not supported. This property is always set to **false**.

Xbox 360 Specific Information

Album information is not supported. This property is always set to **false**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasArt { get; }
```

Property Value

true if this **Album** has associated album art, **false** otherwise.

See Also

Reference

[Album.GetAlbumArt Method](#)

[Album.GetThumbnail Method](#)

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Name Property

Gets the name of the [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **Album**.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Album.Songs Property

Gets a [SongCollection](#) that contains the songs on the album.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SongCollection Songs { get; }
```

Property Value

[SongCollection](#) containing the songs on this album.

See Also

Reference

[Album Class](#)

[Album Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AlbumCollection Class

A collection of albums in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class AlbumCollection : IEnumerable<Album>, IEnumerable
```

Remarks

The [AlbumCollection](#) class provides access to albums in the device's media library.

Use the [MediaLibrary.Albums](#) property to obtain a collection of all albums in the media library, the [Artist.Albums](#) property to obtain a collection of albums associated with a particular artist, and the [Genre.Albums](#) property to obtain a collection of albums associated with a particular genre.

The **AlbumCollection** does not immediately instantiate instances of all albums in the collection. Instead, individual [Album](#) objects are created each time an album is accessed through the collection's [Item](#) indexer. See the [Remarks](#) section of [Item](#) for the resource implications of maintaining references to multiple [Album](#) objects.

See Also

Reference

[MediaLibrary.Albums Property](#)

[Artist.Albums Property](#)

[Genre.Albums Property](#)

[Album Class](#)

[AlbumCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








AlbumCollection Members

The following tables list the members exposed by the AlbumCollection type.



Public Properties

	Name	Description
	Count	Gets the number of Album objects in the AlbumCollection.
	Item	Gets the Album at the specified index in the AlbumCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the AlbumCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the AlbumCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[AlbumCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

AlbumCollection Methods


Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the AlbumCollection.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the AlbumCollection.
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[AlbumCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

AlbumCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[AlbumCollection Class](#)

[AlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AlbumCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [AlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[AlbumCollection Class](#)

[AlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AlbumCollection.GetEnumerator Method

Returns an enumerator that iterates through the [AlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Album> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **AlbumCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[AlbumCollection Class](#)

[AlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[AlbumCollection Class](#)



[AlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AlbumCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of Album objects in the AlbumCollection.
	Item	Gets the Album at the specified index in the AlbumCollection.

See Also

Reference

[AlbumCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

AlbumCollection.Count Property

Gets the number of [Album](#) objects in the [AlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [Album](#) objects in this **AlbumCollection**.

See Also

Reference

[AlbumCollection Class](#)

[AlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AlbumCollection.Item Property

Gets the [Album](#) at the specified index in the [AlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Album this [
    int index
] { get; }
```

Property Value

A new [Album](#) representing the album at the specified index in this **AlbumCollection**.

Remarks

Each call returns a new [Album](#) instance.

Album resources such as art and thumbnails are cached in each **Album** instance. Retrieving and maintaining references to multiple instances of the same album, and accessing art or thumbnails on each of those instances, creates multiple copies of the art or thumbnails. Because large numbers of album art textures can consume considerable memory, it is not advisable to hold a large number of **Album** objects on which [GetAlbumArt](#) has been called.

See Also

Reference

[Album.GetAlbumArt Method](#)

[AlbumCollection Class](#)

[AlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist Class

Provides access to artist information in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Artist : IEquatable<Artist>
```

Remarks

The [Artist](#) class provides information about an artist, including the artist's [Name](#), [Albums](#), and [Songs](#).

You can obtain an **Artist** object through the [ArtistCollection.Item](#) indexer and the [Album.Artist](#) and [Song.Artist](#) properties.

See Also

Reference

[ArtistCollection.Item](#) Property

[Album.Artist](#) Property

[Song.Artist](#) Property

[Artist](#) Members




[Microsoft.Xna.Framework.Media](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









Artist Members

The following tables list the members exposed by the Artist type.



Public Properties

	Name	Description
	Albums	Gets the AlbumCollection for the Artist.
	Name	Gets the name of the Artist.
	Songs	Gets the SongCollection for the Artist.

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Artist are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether the specified Artist instances are equal.
	Op_Inequality	Determines whether the specified Artist instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Artist.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Artist is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[Artist Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Artist Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Artist are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether the specified Artist instances are equal.
	 op_Inequality	Determines whether the specified Artist instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Artist.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Artist is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Artist Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Artist.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.Equals Method

Determines whether two instances of [Artist](#) are equal.

Overload List

Name	Description
Artist.Equals (Artist)	Determines whether the specified Artist is equal to this Artist .
Artist.Equals (Object)	Determines whether the specified Object is equal to this Artist .
Artist.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Artist.Equals Method (Artist)

Determines whether the specified [Artist](#) is equal to this **Artist**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Artist other  
)
```

Parameters

other

Artist to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [Artist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [Artist](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.op_Equality Method

Determines whether the specified [Artist](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Artist first,  
    Artist second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.op_Inequality Method

Determines whether the specified [Artist](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Artist first,  
    Artist second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.ToString Method

Returns a [String](#) representation of the [Artist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[Artist Class](#)




[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist Properties

Public Properties

	Name	Description
	Albums	Gets the AlbumCollection for the Artist.
	Name	Gets the name of the Artist.
	Songs	Gets the SongCollection for the Artist.

See Also

Reference

[Artist Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Artist.Albums Property

Gets the [AlbumCollection](#) for the [Artist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AlbumCollection Albums { get; }
```

Property Value

[AlbumCollection](#) for this **Artist**.

Remarks

The returned [AlbumCollection](#) contains all albums in the device's media library that are associated with this **Artist**.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

Artist.Name Property

Gets the name of the [Artist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **Artist**.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Artist.Songs Property

Gets the [SongCollection](#) for the [Artist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SongCollection Songs { get; }
```

Property Value

[SongCollection](#) for this **Artist**.

Remarks

The returned [SongCollection](#) contains all songs in the device's media library that are associated with this **Artist**.

See Also

Reference

[Artist Class](#)

[Artist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ArtistCollection Class

The collection of all artists in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class ArtistCollection : IEnumerable<Artist>, IEnumerable
```

Remarks

The [ArtistCollection](#) class provides access to all artists in the device's media library.

Use the [MediaLibrary.Artists](#) property to obtain the artist collection.

The **ArtistCollection** does not immediately instantiate instances of all artists in the collection. Instead, individual [Artist](#) objects are created each time an artist is accessed through the collection's [Item](#) indexer.

See Also

Reference

[MediaLibrary.Artists Property](#)

[Artist Class](#)

[ArtistCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








ArtistCollection Members

The following tables list the members exposed by the ArtistCollection type.



Public Properties

	Name	Description
	Count	Gets the number of Artist objects in the ArtistCollection.
	Item	Gets the Artist at the specified index in the ArtistCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the ArtistCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the ArtistCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[ArtistCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

ArtistCollection Methods


Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the ArtistCollection.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the ArtistCollection.
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[ArtistCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

ArtistCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[ArtistCollection Class](#)

[ArtistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ArtistCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [ArtistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[ArtistCollection Class](#)

[ArtistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ArtistCollection.GetEnumerator Method

Returns an enumerator that iterates through the [ArtistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Artist> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **ArtistCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[ArtistCollection Class](#)

[ArtistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[ArtistCollection Class](#)



[ArtistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ArtistCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of Artist objects in the ArtistCollection.
	Item	Gets the Artist at the specified index in the ArtistCollection.

See Also

Reference

[ArtistCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

ArtistCollection.Count Property

Gets the number of [Artist](#) objects in the [ArtistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [Artist](#) objects in this **ArtistCollection**.

See Also

Reference

[ArtistCollection Class](#)

[ArtistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

ArtistCollection.Item Property

Gets the [Artist](#) at the specified index in the [ArtistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Artist this [
    int index
] { get; }
```

Property Value

A new [Artist](#) representing the artist at the specified index in this **ArtistCollection**.

Remarks

Each call returns a new [Artist](#) instance.

Artist resources such as the song and album collections for that artist are cached in each **Artist** instance. Retrieving and maintaining references to multiple instances of the same artist, and accessing the [Songs](#) or [Albums](#) collections on each of those instances, creates multiple copies of the song or album collections.

See Also

Reference

[Artist Class](#)

[ArtistCollection Class](#)

[ArtistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre Class

Provides access to genre information in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Genre : IEquatable<Genre>
```

Remarks

The [Genre](#) class provides information about a genre, including the genre's [Name](#), and the [Albums](#) and [Songs](#) in that genre that are on the device.

You can obtain a **Genre** object through the [GenreCollection.Item](#) indexer and the [Album.Genre](#) and [Song.Genre](#) properties.

See Also

Reference

[GenreCollection.Item](#) Property

[Album.Genre](#) Property

[Song.Genre](#) Property

[Genre](#) Members




[Microsoft.Xna.Framework.Media](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









Genre Members

The following tables list the members exposed by the Genre type.



Public Properties

	Name	Description
	Albums	Gets the AlbumCollection for the Genre.
	Name	Gets the name of the Genre.
	Songs	Gets the SongCollection for the Genre.

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Genre are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether the specified Genre instances are equal.
	op_Inequality	Determines whether the specified Genre instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Genre.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Genre is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[Genre Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Genre Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Genre are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether the specified Genre instances are equal.
	 op_Inequality	Determines whether the specified Genre instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Genre.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Genre is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Genre Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Genre.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.Equals Method

Determines whether two instances of [Genre](#) are equal.

Overload List

Name	Description
Genre.Equals (Genre)	Determines whether the specified Genre is equal to this Genre .
Genre.Equals (Object)	Determines whether the specified Object is equal to this Genre .
Genre.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Genre.Equals Method (Genre)

Determines whether the specified [Genre](#) is equal to this **Genre**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Genre other  
)
```

Parameters

other

Genre to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [Genre](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [Genre](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.op_Equality Method

Determines whether the specified [Genre](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Genre first,  
    Genre second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.op_Inequality Method

Determines whether the specified [Genre](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Genre first,  
    Genre second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.ToString Method

Returns a [String](#) representation of the [Genre](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[Genre Class](#)




[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre Properties

Public Properties

	Name	Description
	Albums	Gets the AlbumCollection for the Genre.
	Name	Gets the name of the Genre.
	Songs	Gets the SongCollection for the Genre.

See Also

Reference

[Genre Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Genre.Albums Property

Gets the [AlbumCollection](#) for the [Genre](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AlbumCollection Albums { get; }
```

Property Value

[AlbumCollection](#) for this **Genre**.

Remarks

The returned [AlbumCollection](#) contains all albums in the device's media library that are associated with this **Genre**.

See Also

Reference

[AlbumCollection Class](#)

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.Name Property

Gets the name of the [Genre](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **Genre**.

See Also

Reference

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Genre.Songs Property

Gets the [SongCollection](#) for the [Genre](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SongCollection Songs { get; }
```

Property Value

[SongCollection](#) for this **Genre**.

Remarks

The returned [SongCollection](#) contains all songs in the device's media library that are associated with this **Genre**.

See Also

Reference

[SongCollection Class](#)

[Genre Class](#)

[Genre Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GenreCollection Class

The collection of all genres in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class GenreCollection : IEnumerable<Genre>, IEnumerable
```

Remarks

The [GenreCollection](#) class provides access to all genres in the device's media library.

Use the [MediaLibrary.Genres](#) property to obtain the genre collection.

The **GenreCollection** does not immediately instantiate instances of all genres in the collection. Instead, individual [Genre](#) objects are created each time a genre is accessed through the collection's [Item](#) indexer.

See Also

Reference

[MediaLibrary.Genres Property](#)

[Genre Class](#)

[GenreCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








GenreCollection Members

The following tables list the members exposed by the GenreCollection type.



Public Properties

	Name	Description
	Count	Gets the number of Genre objects in the GenreCollection.
	Item	Gets the Genre at the specified index in the GenreCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the GenreCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the GenreCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[GenreCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

GenreCollection Methods


Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the GenreCollection.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the GenreCollection.
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[GenreCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

GenreCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[GenreCollection Class](#)

[GenreCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GenreCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [GenreCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors](#), [Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[GenreCollection Class](#)

[GenreCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GenreCollection.GetEnumerator Method

Returns an enumerator that iterates through the [GenreCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Genre> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **GenreCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[GenreCollection Class](#)

[GenreCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[GenreCollection Class](#)



[GenreCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GenreCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of Genre objects in the GenreCollection.
	Item	Gets the Genre at the specified index in the GenreCollection.

See Also

Reference

[GenreCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

GenreCollection.Count Property

Gets the number of [Genre](#) objects in the [GenreCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [Genre](#) objects in this **GenreCollection**.

See Also

Reference

[GenreCollection Class](#)

[GenreCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GenreCollection.Item Property

Gets the [Genre](#) at the specified index in the [GenreCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Genre this [
    int index
] { get; }
```

Property Value

A new [Genre](#) representing the genre at the specified index in this **GenreCollection**.

Remarks

Each call returns a new [Genre](#) instance.

Genre resources such as the song and album collections for that genre are cached in each **Genre** instance. Retrieving and maintaining references to multiple instances of the same genre, and accessing the [Songs](#) or [Albums](#) collections on each of those instances, creates multiple copies of the song or album collections.

See Also

Reference

[Genre Class](#)

[GenreCollection Class](#)

[GenreCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary Class

Provides access to songs, playlists, and pictures in the device's media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class MediaLibrary
```

Remarks

[MediaLibrary](#) provides the following properties that return media collections: [Albums](#), [Artists](#), [Genres](#), [Pictures](#), [Playlists](#), and [Songs](#). Each property returns a collection object that can be enumerated and indexed. The collection object represents all media of that type in the device's media library.

The media collections do not retrieve or instantiate all media objects immediately. Instead, when needed, you can use the collections later to retrieve individual media objects, such as [Song](#) and [Artist](#) objects.

On Windows, **MediaLibrary** cannot find any songs unless the Windows Media Player previously found songs on the system. That means that Windows Media Player must first search the system for music before any songs can be accessed through **MediaLibrary**.

Example

The following sample fragment demonstrates how to use **MediaLibrary** to retrieve and play a song.

C#

```
using System;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;

namespace ZuneSimpleSongPlay
{
    static class Program
    {
        static void Main(string[] args)
        {
            MediaLibrary library = new MediaLibrary();
            SongCollection songs = library.Songs;
            Song song = songs[0];
            MediaPlayer.Play(song);

            while (!GamePad.GetState(PlayerIndex.One).IsButtonDown(Buttons.Back))
            {
                ;
            }
        }
    }
}
```

Note

You may get an exception when you try to play DRM protected music.

See Also

Tasks

[How To: Show Pictures](#)

Reference

[MediaLibrary Members](#)


[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









MediaLibrary Members

The following tables list the members exposed by the MediaLibrary type.







Public Constructors

Name	Description
 MediaLibrary	Overloaded. Initializes a new instance of this class.



Public Properties

Name	Description
 Albums	Gets the AlbumCollection that contains all albums in the media library.
 Artists	Gets the ArtistCollection that contains all artists in the media library.
 Genres	Gets the GenreCollection that contains all genres in the media library.
 MediaSource	Gets the MediaSource with which this media library was constructed.
 Pictures	Gets the PictureCollection that contains all pictures in the media library.
 Playlists	Gets the PlaylistCollection that contains all playlists in the media library.
 RootPictureAlbum	Gets the root PictureAlbum for all pictures in the media library.
 Songs	Gets the SongCollection that contains all songs in the media library.

Public Methods

Name	Description
 Dispose	Releases the resources used by the MediaLibrary .
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before the MediaLibrary is reclaimed by garbage collection.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MediaLibrary Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaLibrary Constructor

Initializes a new instance of this class.

Overload List

Name	Description
MediaLibrary ()	Initializes a new instance of the MediaLibrary class.
MediaLibrary (MediaSource)	Initializes a new instance of the MediaLibrary class, using a specific media source to create the new media library.

See Also

Reference

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[MediaSource Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaLibrary Constructor ()

Initializes a new instance of the [MediaLibrary](#) class.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MediaLibrary ()
```

See Also

Reference

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary Constructor (MediaSource)

Initializes a new instance of the [MediaLibrary](#) class, using a specific media source to create the new media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MediaLibrary (  
    MediaSource mediaSource  
)
```

Parameters

mediaSource

A media source that will be the source for the media library.

Example

The following example demonstrates creating a new [MediaLibrary](#) class using a [MediaSource](#).

C#

```
GraphicsDevice.Clear(Color.CornflowerBlue);  
Vector2 pos = Vector2.Zero;  
SpriteFont font = Content.Load<SpriteFont>("Font");  
  
spriteBatch.Begin();  
spriteBatch.DrawString(font, "media library stats:",  
    pos, Color.White);  
  
ICollection<MediaSource> mediaSources =  
    MediaSource.GetAvailableMediaSources();  
foreach (MediaSource source in mediaSources)  
{  
    MediaLibrary ml = new MediaLibrary(source);  
  
    // display some data about this media library  
    pos.Y += font.LineSpacing;  
    spriteBatch.DrawString(font, "Album count: " + ml.Albums.Count,  
        pos, Color.White);  
}  
  
spriteBatch.End();
```

See Also

Reference

[MediaLibrary Class](#)

[MediaSource Class](#)







[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary Methods

Public Methods

	Name	Description
	Dispose	Releases the resources used by the MediaLibrary .
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the MediaLibrary is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MediaLibrary Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaLibrary.Dispose Method

Releases the resources used by the [MediaLibrary](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

Call **Dispose** only when you are finished using the **MediaLibrary**. The **Dispose** method leaves the **MediaLibrary** in an unusable state. After calling **Dispose**, you must release all references to the **MediaLibrary** in order for the garbage collector to reclaim the memory that the **MediaLibrary** was occupying. For more information, see [Cleaning Up Unmanaged Resources](#) and [Implementing a Dispose Method](#).

See Also

Reference

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [MediaLibrary](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[MediaLibrary Class](#)









[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary Properties

Public Properties

	Name	Description
	Albums	Gets the AlbumCollection that contains all albums in the media library.
	Artists	Gets the ArtistCollection that contains all artists in the media library.
	Genres	Gets the GenreCollection that contains all genres in the media library.
	MediaSource	Gets the MediaSource with which this media library was constructed.
	Pictures	Gets the PictureCollection that contains all pictures in the media library.
	Playlists	Gets the PlaylistCollection that contains all playlists in the media library.
	RootPictureAlbum	Gets the root PictureAlbum for all pictures in the media library.
	Songs	Gets the SongCollection that contains all songs in the media library.

See Also

Reference

[MediaLibrary Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaLibrary.Albums Property

Gets the [AlbumCollection](#) that contains all albums in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public AlbumCollection Albums { get; }
```

Property Value

[AlbumCollection](#) that contains all albums in the device's media library.

See Also

Reference

[AlbumCollection Class](#)

[Album Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.Artists Property

Gets the [ArtistCollection](#) that contains all artists in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ArtistCollection Artists { get; }
```

Property Value

[ArtistCollection](#) that contains all artists in the device's media library.

See Also

Reference

[ArtistCollection Class](#)

[Artist Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.Genres Property

Gets the [GenreCollection](#) that contains all genres in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GenreCollection Genres { get; }
```

Property Value

[GenreCollection](#) that contains all genres in the device's media library.

See Also

Reference

[GenreCollection Class](#)

[Genre Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.MediaSource Property

Gets the [MediaSource](#) with which this media library was constructed.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MediaSource MediaSource { get; }
```

Property Value

The [MediaSource](#) with which this media library was constructed.

See Also

Reference

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[MediaSource Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.Pictures Property

Gets the [PictureCollection](#) that contains all pictures in the media library.

Windows Specific Information

Picture collections are not supported. This property always returns an empty collection.

Xbox 360 Specific Information

Picture collections are not supported. This property always returns an empty collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureCollection Pictures { get; }
```

Property Value

[PictureCollection](#) that contains all pictures in the device's media library.

See Also

Reference

[PictureCollection Class](#)

[Picture Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.Playlists Property

Gets the [PlaylistCollection](#) that contains all playlists in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PlaylistCollection Playlists { get; }
```

Property Value

[PlaylistCollection](#) that contains all playlists in the device's media library.

See Also

Reference

[PlaylistCollection Class](#)

[Playlist Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.RootPictureAlbum Property

Gets the root [PictureAlbum](#) for all pictures in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureAlbum RootPictureAlbum { get; }
```

Property Value

The root [PictureAlbum](#) for all pictures in the device's media library.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbumCollection Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaLibrary.Songs Property

Gets the [SongCollection](#) that contains all songs in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SongCollection Songs { get; }
```

Property Value

[SongCollection](#) that contains all songs in the device's media library.

See Also

Reference

[SongCollection Class](#)

[Song Class](#)

[MediaLibrary Class](#)

[MediaLibrary Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer Class

Provides methods and properties to play, pause, resume, and stop songs. [MediaPlayer](#) also exposes shuffle, repeat, volume, play position, and visualization capabilities.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static class MediaPlayer
```

Remarks

MediaPlayer provides methods and properties for playing songs in the media library.

To control song playback, use the [Play](#), [Pause](#), [Stop](#), and [Resume](#) methods. The [MoveNext](#) and [MovePrevious](#) methods move to the next or previous song in the queue. They operate as if the queue was circular. That is, [MoveNext](#), when the last song is playing, moves to the first song. [MovePrevious](#), when the first song is playing, moves to the last song.

To get and set playback options, use the [IsMuted](#), [IsRepeating](#), [IsShuffled](#), [PlayPosition](#), and [Volume](#) properties.

To obtain visualization data for the currently playing song, call [GetVisualizationData](#).

MediaPlayer is a static class, so you do not have to use [new](#) to instantiate an instance of the class. Instead, call methods and properties directly on the class.

See Also

Tasks

[How To: Play a Song](#)

Reference

[MediaPlayer.Play Method](#)

[Song Class](#)

[MediaPlayer Members](#)










[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




















MediaPlayer Members

The following tables list the members exposed by the MediaPlayer type.



Public Properties

	Name	Description
	GameHasControl	Determines whether the game has control of the background music.
	IsMuted	Gets or set the muted setting for the media player.
	IsRepeating	Gets or sets the repeat setting for the media player.
	IsShuffled	Gets or sets the shuffle setting for the media player.
	IsVisualizationEnabled	Gets or sets the visualization enabled setting for the media player.
	PlayPosition	Gets the play position within the currently playing song.
	Queue	Gets the media playback queue, MediaQueue .
	State	Gets the media playback state, MediaState .
	Volume	Gets or sets the media player volume.





Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	 GetVisualizationData	Retrieves visualization (frequency and sample) data for the currently-playing song.
	 MoveNext	Moves to the next song in the queue of playing songs.
	 MovePrevious	Moves to the previous song in the queue of playing songs.
	 Pause	Pauses the currently playing song.
	 Play	Overloaded. Plays a song or collection of songs.
	ReferenceEquals	(Inherited from Object .)
	 Resume	Resumes a paused song.
	 Stop	Stops playing a song.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Public Events

	Name	Description
	 ActiveSongChanged	Raised when the active song changes due to active playback or due to explicit calls to the MoveNext or MovePrevious methods.
	 MediaStateChanged	Raised when the media player play state changes.

See Also




















Reference

[MediaPlayer Class](#)





[Microsoft.Xna.Framework.Media Namespace](#)

MediaPlayer Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
 	GetVisualizationData	Retrieves visualization (frequency and sample) data for the currently-playing song.
 	MoveNext	Moves to the next song in the queue of playing songs.
 	MovePrevious	Moves to the previous song in the queue of playing songs.
 	Pause	Pauses the currently playing song.
 	Play	Overloaded. Plays a song or collection of songs.
	ReferenceEquals	(Inherited from Object .)
 	Resume	Resumes a paused song.
 	Stop	Stops playing a song.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
 	Finalize	(Inherited from Object .)
 	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MediaPlayer Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaPlayer.GetVisualizationData Method

Retrieves visualization (frequency and sample) data for the currently-playing song.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void GetVisualizationData (
    VisualizationData visualizationData
)
```

Parameters

visualizationData

Visualization (frequency and sample) data for the currently playing song.

Remarks

Visualization data can be used to create *visualizers*, which are graphical representations of sound. For example, a simple visualizer might be an equalizer spectrum display that shows the current power level for each frequency band.

The frequency data is a collection of float values. Each element corresponds to a frequency band, ranging from 20Hz to 20KHz. In the collection, the distribution of bands from 20Hz to 20KHz is logarithmic, not linear. This means that elements at the lower end of the spectrum represent a smaller frequency range than those at the upper end of the spectrum.

Each value in the frequency collection is a normalized float value from 0.0f to 1.0f, and is the logarithmic scaled power level for that frequency band. Frequency is perceived as pitch.

Information returned from the **GetVisualizationData** method can determine if a sound is more of a higher pitch, or more of a lower pitch. This information can be useful when determining how the sounds might drive game play.

The Complete Sample

The code in the topic shows you the technique. You can download a complete code sample for this topic, including full source code and any additional supporting files required by the sample.

[Download Spectrum.zip](#).

Example

The following example demonstrates using frequency and sample data to display the frequency power spectrum and waveform of the currently playing song. The program display is in landscape mode.

C#

```
    if (MediaPlayer.State == MediaState.Playing)
    {
        MediaPlayer.GetVisualizationData(visData);
    }
    ...
    spriteBatch.Begin();
    for (int y = 0; y < arraySize; y++)
    {
        // Draw frequency spectrum display.
        spriteBatch.Draw(
            line,
            new Vector2((float)((-1.0 + visData.Frequencies[y]) * width + 1), y),
            Color.White);

        // Draw waveform from samples.
        spriteBatch.Draw(
            line,
            new Vector2((float)(visData.Samples[y] * width / 4 + width * 3 / 4), y),
            sampleLine,
            Color.White);
    }
```



```
spriteBatch.End();
```

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.MoveNext Method

Moves to the next song in the queue of playing songs.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void MoveNext ()
```

Remarks

If the current song is the last song in the queue, **MoveNext** moves to the first song in the queue.

MoveNext does not alter the current media state (playing, paused, or stopped).

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.MovePrevious Method

Moves to the previous song in the queue of playing songs.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void MovePrevious ()
```

Remarks

If the current song is the first song in the queue, **MovePrevious** moves to the last song in the queue.

MovePrevious does not alter the current media state (playing, paused, or stopped).

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Pause Method

Pauses the currently playing song.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Pause ()
```

See Also

Reference

[MediaPlayer.Play Method](#)

[MediaPlayer.Resume Method](#)

[MediaPlayer.Stop Method](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Play Method

Plays a song or collection of songs.

Overload List

Name	Description
MediaPlayer.Play (Song)	Plays a Song .
MediaPlayer.Play (SongCollection)	Plays a SongCollection .
MediaPlayer.Play (SongCollection, Int32)	Plays a SongCollection , starting with the Song at the specified index.

See Also

Tasks

[How To: Play a Song](#)

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaPlayer.Play Method (Song)

Plays a [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Play (  
    Song song  
)
```

Parameters

song

[Song](#) to play.

Exceptions

Exception type	Condition
ArgumentNullException	<i>song</i> is null .

Remarks

Play clears the current playback queue, and then queues up the specified song for playback. Playback starts immediately at the beginning of the song.

See Also

Tasks

[How To: Play a Song](#)

Reference

[MediaPlayer.Play Method](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Play Method (SongCollection)

Plays a [SongCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Play (  
    SongCollection songs  
)
```

Parameters

songs

[SongCollection](#) to play.

Exceptions

Exception type	Condition
ArgumentException	Play cannot be used when the song collection count is zero.
ArgumentNullException	<i>songs</i> is null .

Remarks

Play clears the current playback queue, and then queues up the specified song collection for playback. It begins with the first song in the collection. Playback starts immediately at the beginning of the song.

See Also

Tasks

[How To: Play a Song](#)

Reference

[MediaPlayer.Play Method](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Play Method (SongCollection, Int32)

Plays a [SongCollection](#), starting with the [Song](#) at the specified index.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Play (  
    SongCollection songs,  
    int index  
)
```

Parameters

songs

[SongCollection](#) to play.

index

Index of the song in the collection at which playback should begin.

Exceptions

Exception type	Condition
ArgumentException	Play cannot be used when the song collection count is zero.
ArgumentNullException	<i>songs</i> is null .
ArgumentOutOfRangeException	<i>index</i> must be valid for the current collection.

Remarks

Play clears the current playback queue, and then queues up the specified song collection for playback. It begins with the song at the specified index in the collection. Playback starts immediately at the beginning of the song.

See Also

Tasks

[How To: Play a Song](#)

Reference

[MediaPlayer.Play Method](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Resume Method

Resumes a paused song.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Resume ()
```

See Also

Reference

[MediaPlayer.Play Method](#)

[MediaPlayer.Pause Method](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Stop Method

Stops playing a song.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static void Stop ()
```

See Also

Reference

[MediaPlayer.Play Method](#)

[MediaPlayer.Pause Method](#)

[MediaPlayer Class](#)










[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer Properties

Public Properties

	Name	Description
 S	GameHasControl	Determines whether the game has control of the background music.
 S	IsMuted	Gets or set the muted setting for the media player.
 S	IsRepeating	Gets or sets the repeat setting for the media player.
 S	IsShuffled	Gets or sets the shuffle setting for the media player.
 S	IsVisualizationEnabled	Gets or sets the visualization enabled setting for the media player.
 S	PlayPosition	Gets the play position within the currently playing song.
 S	Queue	Gets the media playback queue, MediaQueue .
 S	State	Gets the media playback state, MediaState .
 S	Volume	Gets or sets the media player volume.

See Also

Reference

[MediaPlayer Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaPlayer.GameHasControl Property

Determines whether the game has control of the background music.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool GameHasControl { get; }
```

Property Value

true if the game has control of the background music; otherwise, **false**.

Remarks

A gamer can play his or her own music as the background to your game by using the Xbox 360 dashboard. If the game is currently playing custom background music (specified by the gamer using the Xbox 360 dashboard), calls to [Play](#), [Stop](#), [Pause](#), [Resume](#), [MoveNext](#), and [MovePrevious](#) have no effect.

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.IsMuted Property

Gets or set the muted setting for the media player.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsMuted { get; set; }
```

Property Value

true if sound is muted; **false** otherwise.

See Also

Reference

[MediaPlayer.Volume Property](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.IsRepeating Property

Gets or sets the repeat setting for the media player.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsRepeating { get; set; }
```

Property Value

true if the current play queue is set to repeat; **false** otherwise.

Remarks

When **IsRepeating** is **true**, the playback queue will begin playing again after all songs in the queue have been played.

See Also

Reference

[MediaPlayer.IsShuffled Property](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.IsShuffled Property

Gets or sets the shuffle setting for the media player.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsShuffled { get; set; }
```

Property Value

true if the current play queue is set to shuffled; **false** otherwise.

Remarks

When **IsShuffled** is **true**, songs in the playback queue are played in random order rather than from first to last.

See Also

Reference

[MediaPlayer.IsRepeating Property](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.IsVisualizationEnabled Property

Gets or sets the visualization enabled setting for the media player.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool IsVisualizationEnabled { get; set; }
```

Property Value

true if visualization is enabled; **false** otherwise.

RemarksWhen **IsVisualizationEnabled** is **true**, visualization data is enabled and [GetVisualizationData](#) will return visualization data for the currently playing song.

See Also

Reference

[MediaPlayer.GetVisualizationData Method](#)

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.PlayPosition Property

Gets the play position within the currently playing song.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static TimeSpan PlayPosition { get; }
```

Property Value

Play position within the current song, as a [TimeSpan](#) structure.

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Queue Property

Gets the media playback queue, [MediaQueue](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static MediaQueue Queue { get; }
```

Property Value

The media playback queue, [MediaQueue](#).

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.State Property

Gets the media playback state, [MediaState](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static MediaState State { get; }
```

Property Value

The media playback state, [MediaState](#).

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.Volume Property

Gets or sets the media player volume.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static float Volume { get; set; }
```

Property Value

Media player volume, from 0.0f (silence) to 1.0f (full volume relative to the current device volume).

Remarks

Volume adjustment is based on a decibel, not multiplicative, scale. For example, when the device volume is half of maximum (about **7** in the Zune user interface), setting **Volume** to 0.6f or less is silent or nearly so, not volume **4** as you would expect from a multiplicative adjustment.

Setting **Volume** to 0.0 subtracts 96 dB from the volume. Setting **Volume** to 1.0 subtracts 0 dB from the volume. Values in between 0.0f and 1.0f subtract dB from the volume proportionally.

See Also

Reference

[MediaPlayer Class](#)





[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer Events

Public Events

	Name	Description
 	ActiveSongChanged	Raised when the active song changes due to active playback or due to explicit calls to the MoveNext or MovePrevious methods.
 	MediaStateChanged	Raised when the media player play state changes.

See Also

Reference

[MediaPlayer Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaPlayer.ActiveSongChanged Event

Raised when the active song changes due to active playback or due to explicit calls to the [MoveNext](#) or [MovePrevious](#) methods.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler ActiveSongChanged
```

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[MoveNext](#)

[MovePrevious](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaPlayer.MediaStateChanged Event

Raised when the media player play state changes.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler MediaStateChanged
```

See Also

Reference

[MediaPlayer Class](#)

[MediaPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaQueue Class

Provides methods and properties to access and control the queue of playing songs.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class MediaQueue
```

Remarks

[MediaQueue](#) provides properties and methods that query and control the queue of playing songs. The [ActiveSong](#) property returns the currently playing [Song](#). [ActiveSongIndex](#) gets or set the queue index of the currently playing song. [Count](#) returns the number of songs in the queue. The [Item](#) indexer returns the [Song](#) at the specified index in the queue.

MediaQueue is a read-only queue of songs. With **MediaQueue**, you can control which song is playing in the queue, but you cannot add or remove songs from the queue. Either [MediaPlayer.Play](#) or the songs already queued up when the game starts determine the songs that are in the queue of playing songs.

Note

All of the collections, playlists, and queues returned by methods and properties in the [Microsoft.Xna.Framework.Media](#) name space are immutable. You cannot add or remove objects from those collections or playlists. To create a custom 'playlist' of songs, games must maintain their own list of songs to play, and play those songs one at a time by calling [MediaPlayer.Play](#).

See Also

Reference

[MediaPlayer.Play Method](#)

[MediaQueue Members](#)





[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






MediaQueue Members

The following tables list the members exposed by the MediaQueue type.



Public Properties

Name	Description
 ActiveSong	Gets the current Song in the queue of playing songs.
 ActiveSongIndex	Gets or sets the index of the current (active) song in the queue of playing songs.
 Count	Gets the count of songs in the MediaQueue.
 Item	Gets the Song at the specified index in the MediaQueue.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also





Reference

[MediaQueue Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaQueue Properties

Public Properties

	Name	Description
	ActiveSong	Gets the current Song in the queue of playing songs.
	ActiveSongIndex	Gets or sets the index of the current (active) song in the queue of playing songs.
	Count	Gets the count of songs in the MediaQueue.
	Item	Gets the Song at the specified index in the MediaQueue.

See Also

Reference

[MediaQueue Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaQueue.ActiveSong Property

Gets the current [Song](#) in the queue of playing songs.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Song ActiveSong { get; }
```

Property Value

Current [Song](#) in the queue of playing songs.

Remarks

Retrieving this property is expensive on Xbox, so it should not be called every frame.

See Also

Reference

[MediaQueue.ActiveSongIndex Property](#)

[MediaQueue Class](#)

[MediaQueue Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaQueue.ActiveSongIndex Property

Gets or sets the index of the current (active) song in the queue of playing songs.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ActiveSongIndex { get; set; }
```

Property Value

Index of the current song in the queue of playing songs.

Remarks

Changing the active song index does not alter the current media state (playing, paused, or stopped).

See Also

Reference

[MediaQueue.ActiveSong Property](#)

[MediaQueue Class](#)

[MediaQueue Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaQueue.Count Property

Gets the count of songs in the [MediaQueue](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Count of songs in this **MediaQueue**.

See Also

Reference

[MediaQueue.ActiveSongIndex Property](#)

[MediaQueue Class](#)

[MediaQueue Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaQueue.Item Property

Gets the [Song](#) at the specified index in the [MediaQueue](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Song this [
    int index
] { get; }
```

Property Value

A new [Song](#) representing the song at the specified index in this **MediaQueue**.

Remarks

Each call returns a new [Song](#) instance.

See Also

Reference

[MediaQueue.ActiveSongIndex Property](#)

[MediaQueue Class](#)

[MediaQueue Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaSource Class

Provides methods and properties to access the source or sources from which the media will be read.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class MediaSource
```

Remarks

[MediaSource](#) provides access to the source or sources from which the media will be read. A source can be either the local device, or a device connected through Windows Media Connect. On Windows and Zune, the only available **MediaSource** is the local device. On Xbox 360, a **MediaSource** can either be the local device or a device connected through Windows Media Connect. Windows Media Connect is software that lets you connect your Xbox 360 console to a computer running Microsoft Windows. Connecting this way enables you to view media on the connected computer in the Xbox 360 dashboard.

See Also

Reference

[MediaSource Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








MediaSource Members

The following tables list the members exposed by the MediaSource type.



Public Properties

	Name	Description
	MediaSourceType	Gets the MediaSourceType of this media source.
	Name	Gets the name of this media source.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	 GetAvailableMediaSources	Gets the available media sources.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns the name of this media source.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[MediaSource Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

MediaSource Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
 	GetAvailableMediaSources	Gets the available media sources.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns the name of this media source.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MediaSource Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaSource.GetAvailableMediaSources Method

Gets the available media sources.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IList<MediaSource> GetAvailableMediaSources ()
```

Return Value

The available media sources.

Remarks Gets the available media sources with which a media library can be constructed. On Zune and Windows, this method will always return a single media source: the local device. On Xbox 360, this method can return multiple sources, depending on how many computers are connected to the Xbox 360 through Windows Media Connect.

Example

The following example demonstrates how the **GetAvailableMediaSources** method gets and displays the current media sources.

C#

```
GraphicsDevice.Clear(Color.CornflowerBlue);
Vector2 pos = Vector2.Zero;
SpriteFont font = Content.Load<SpriteFont>("Font");

spriteBatch.Begin();
spriteBatch.DrawString(font, "media source(s):", pos, Color.White);

ICollection<MediaSource> mediaSources =
    MediaSource.GetAvailableMediaSources();
foreach (MediaSource source in mediaSources)
{
    pos.Y += font.LineSpacing;
    spriteBatch.DrawString(font, source.Name, pos, Color.White);
}

spriteBatch.End();
```

See Also

Reference

[MediaSource Class](#)

[MediaSource Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaSource.ToString Method

Returns the name of this media source.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

The name of this media source.

See Also

Reference

[MediaSource Class](#)



[MediaSource Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaSource Properties

Public Properties

	Name	Description
	MediaSourceType	Gets the MediaSourceType of this media source.
	Name	Gets the name of this media source.

See Also

Reference

[MediaSource Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

MediaSource.MediaType Property

Gets the [MediaType](#) of this media source.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MediaType MediaType { get; }
```

Property Value

The [MediaType](#) of this media source.

See Also

Reference

[MediaSource Class](#)

[MediaSource Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaSource.Name Property

Gets the name of this media source.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The media source name.

See Also

Reference

[MediaSource Class](#)

[MediaSource Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaSourceType Enumeration

Type of the media source.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum MediaSourceType
```

Members

Member name	Description
LocalDevice	A local device.
WindowsMediaConnect	A Windows Media Connect device.

Remarks

Indicates the type of the current media source. The type can be either a local device, or a device connected through Windows Media Connect. On Windows and Zune, the only available media source type is the local device. On Xbox 360, a media source can be either the local device or a device connected through Windows Media Connect.

The [MediaSource.MediaSourceType](#) property returns a **MediaSourceType** value.

See Also

Reference

[MediaSource.MediaSourceType Property](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

MediaState Enumeration

Media playback state (playing, paused, or stopped).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum MediaState
```

Members

Member name	Description
Paused	Media playback is paused.
Playing	Media is currently playing.
Stopped	Media playback is stopped.

Remarks

The [MediaPlayer.State](#) property returns a **MediaState** value.

See Also

Reference

[MediaPlayer.State Property](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture Class

Provides access to a picture in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Picture : IEquatable<Picture>
```

Remarks

The [Picture](#) class provides information about a picture, including the picture's graphic ([GetTexture](#)), thumbnail ([GetThumbnail](#)), [Name](#), [Date](#), and [Height](#) and [Width](#).

You can obtain a **Picture** object through the [PictureCollection.Item](#) indexer.

See Also

Reference

[PictureCollection.Item](#) Property

[Picture](#) Members






[Microsoft.Xna.Framework.Media](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune











Picture Members

The following tables list the members exposed by the Picture type.



Public Properties

	Name	Description
	Album	Gets the picture album that contains the picture.
	Date	Gets the picture's date.
	Height	Gets the picture's height.
	Name	Gets the name of the Picture.
	Width	Gets the picture's width.

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Picture are equal.
	GetHashCode	Gets the hash code for this instance.
	GetTexture	Gets the picture as a texture.
	GetThumbnail	Gets a thumbnail of the picture as a texture.
	GetType	(Inherited from Object .)
	op_Equality	Determines whether the specified Picture instances are equal.
	op_Inequality	Determines whether the specified Picture instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Picture.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Picture is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also













Reference

[Picture Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Picture Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Picture are equal.
	GetHashCode	Gets the hash code for this instance.
	GetTexture	Gets the picture as a texture.
	GetThumbnail	Gets a thumbnail of the picture as a texture.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether the specified Picture instances are equal.
	 op_Inequality	Determines whether the specified Picture instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Picture.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Picture is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Picture Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Picture.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Equals Method

Determines whether two instances of [Picture](#) are equal.

Overload List

Name	Description
Picture.Equals (Object)	Determines whether the specified Object is equal to this Picture .
Picture.Equals (Picture)	Determines whether the specified Picture is equal to this Picture .
Picture.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Picture.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [Picture](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Equals Method (Picture)

Determines whether the specified [Picture](#) is equal to this **Picture**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Picture other  
)
```

Parameters

other

Picture to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [Picture](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.GetTexture Method

Gets the picture as a texture.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetTexture (  
    IServiceProvider serviceProvider  
)
```

Parameters

serviceProvider

Game service provider, obtained from the [Game.Services](#) property.

Return Value

Texture with the graphical content of the picture.

Remarks

[GameServiceContainer Class](#) implements [IServiceProvider](#), so you can pass the value of the [Game.Services](#) property as the *serviceProvider* parameter.

See Also

Reference

[Picture.GetThumbnail Method](#)

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.GetThumbnail Method

Gets a thumbnail of the picture as a texture.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetThumbnail (  
    IServiceProvider serviceProvider  
)
```

Parameters

serviceProvider

Game service provider, obtained from the [Game.Services](#) property.

Return Value

Texture with a thumbnail of the picture's graphical content.

Remarks

[GameServiceContainer Class](#) implements [IServiceProvider](#), so you can pass the value of the [Game.Services](#) property as the *serviceProvider* parameter.

See Also

Reference

[Picture.GetTexture Method](#)

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.op_Equality Method

Determines whether the specified [Picture](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Picture first,  
    Picture second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.op_Inequality Method

Determines whether the specified [Picture](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Picture first,  
    Picture second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.ToString Method

Returns a [String](#) representation of the [Picture](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[Picture Class](#)






[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture Properties

Public Properties

	Name	Description
	Album	Gets the picture album that contains the picture.
	Date	Gets the picture's date.
	Height	Gets the picture's height.
	Name	Gets the name of the Picture.
	Width	Gets the picture's width.

See Also

Reference

[Picture Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Picture.Album Property

Gets the picture album that contains the picture.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureAlbum Album { get; }
```

Property Value

[PictureAlbum](#) that contains this picture.

See Also

Reference

[PictureAlbum Class](#)

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Date Property

Gets the picture's date.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public DateTime Date { get; }
```

Property Value

Date and time associated with the picture's file, as a [DateTime](#) structure.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Height Property

Gets the picture's height.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

Height of this picture, in pixels.

See Also

Reference

[Picture.Width Property](#)

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Name Property

Gets the name of the [Picture](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **Picture**.

See Also

Reference

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Picture.Width Property

Gets the picture's width.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

Width of this picture, in pixels.

See Also

Reference

[Picture.Height Property](#)

[Picture Class](#)

[Picture Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum Class

Provides access to a picture album in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class PictureAlbum : IEquatable<PictureAlbum>
```

Remarks

The [PictureAlbum](#) class provides information about a picture album, including the picture album's [Name](#), [Pictures](#), and [Parent](#).

A **PictureAlbum** can contain both pictures and other picture albums.

Obtain **PictureAlbum** objects through the [PictureAlbumCollection.Item](#) indexer.

See Also

Reference

[PictureAlbumCollection.Item](#) Property

[PictureAlbum](#) Members





[Microsoft.Xna.Framework.Media](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









PictureAlbum Members

The following tables list the members exposed by the PictureAlbum type.



Public Properties

	Name	Description
	Albums	Gets the collection of picture albums that are contained within the picture album (that is, picture albums that are children of the picture album).
	Name	Gets the name of the PictureAlbum.
	Parent	Gets the parent picture album.
	Pictures	Gets the collection of pictures in this picture album.

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of PictureAlbum are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Sop_Equality	Determines whether the specified PictureAlbum instances are equal.
	Sop_Inequality	Determines whether the specified PictureAlbum instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the PictureAlbum.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the PictureAlbum is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[PictureAlbum Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbum Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of PictureAlbum are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 Op_Equality	Determines whether the specified PictureAlbum instances are equal.
	 Op_Inequality	Determines whether the specified PictureAlbum instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the PictureAlbum.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the PictureAlbum is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PictureAlbum Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbum.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.Equals Method

Determines whether two instances of [PictureAlbum](#) are equal.

Overload List

Name	Description
PictureAlbum.Equals (Object)	Determines whether the specified Object is equal to this PictureAlbum .
PictureAlbum.Equals (PictureAlbum)	Determines whether the specified PictureAlbum is equal to this PictureAlbum .
PictureAlbum.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbum.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [PictureAlbum](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.Equals Method (PictureAlbum)

Determines whether the specified [PictureAlbum](#) is equal to this **PictureAlbum**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    PictureAlbum other  
)
```

Parameters

other

PictureAlbum to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [PictureAlbum](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.op_Equality Method

Determines whether the specified [PictureAlbum](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    PictureAlbum first,  
    PictureAlbum second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.op_Inequality Method

Determines whether the specified [PictureAlbum](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    PictureAlbum first,  
    PictureAlbum second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.ToString Method

Returns a [String](#) representation of the [PictureAlbum](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[PictureAlbum Class](#)





[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum Properties

Public Properties

	Name	Description
	Albums	Gets the collection of picture albums that are contained within the picture album (that is, picture albums that are children of the picture album).
	Name	Gets the name of the PictureAlbum.
	Parent	Gets the parent picture album.
	Pictures	Gets the collection of pictures in this picture album.

See Also

Reference

[PictureAlbum Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbum.Albums Property

Gets the collection of picture albums that are contained within the picture album (that is, picture albums that are children of the picture album).

Windows Specific Information

Picture albums are not supported. This property always returns an empty collection.

Xbox 360 Specific Information

Picture albums are not supported. This property always returns an empty collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureAlbumCollection Albums { get; }
```

Property Value

[PictureAlbumCollection](#) that contains the picture albums that are children of this picture album.

Remarks

A [PictureAlbum](#) can contain pictures as well as other picture albums. This property returns the collection of picture albums that are contained within this picture album, if any. If this picture album has no child picture albums, an empty collection is returned.

See Also

Reference

[PictureAlbumCollection Class](#)

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.Name Property

Gets the name of the [PictureAlbum](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **PictureAlbum**.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.Parent Property

Gets the parent picture album.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureAlbum Parent { get; }
```

Property Value

[PictureAlbum](#) that is the parent of this picture album.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbum.Pictures Property

Gets the collection of pictures in this picture album.

Windows Specific Information

Picture collections are not supported. This property always returns an empty collection.

Xbox 360 Specific Information

Picture collections are not supported. This property always returns an empty collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureCollection Pictures { get; }
```

Property Value

[PictureCollection](#) that contains the pictures in this picture album.

Remarks

A [PictureAlbum](#) can contain pictures as well as other picture albums. This property returns the collection of pictures in this picture album, if any. If this picture album has no pictures, an empty collection is returned.

See Also

Reference

[PictureAlbum Class](#)

[PictureAlbum Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbumCollection Class

A collection of picture albums in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class PictureAlbumCollection : IEnumerable<PictureAlbum>, IEnumerable
```

Remarks

The [PictureAlbumCollection](#) class provides access to picture albums in the device's media library.

Use the [PictureAlbum.Albums](#) property to obtain a collection of picture albums that are contained within (that is, children of) a particular picture album. Use the [MediaLibrary.RootPictureAlbum](#) property to obtain the root (top-most) picture album, which contains both pictures and other picture albums.

The **PictureAlbumCollection** does not immediately instantiate instances of all picture albums in the collection. Instead, individual [PictureAlbum](#) objects are created each time a picture album is accessed through the collection's [Item](#) indexer.

See Also

Reference

[PictureAlbum.Albums Property](#)

[MediaLibrary.RootPictureAlbum Property](#)

[PictureAlbum Class](#)

[PictureAlbumCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








PictureAlbumCollection Members

The following tables list the members exposed by the PictureAlbumCollection type.



Public Properties

	Name	Description
	Count	Gets the number of PictureAlbum objects in the PictureAlbumCollection.
	Item	Gets the PictureAlbum at the specified index in the PictureAlbumCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the PictureAlbumCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the PictureAlbumCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[PictureAlbumCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbumCollection Methods


Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the PictureAlbumCollection.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the PictureAlbumCollection.
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[PictureAlbumCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbumCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[PictureAlbumCollection Class](#)

[PictureAlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbumCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [PictureAlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[PictureAlbumCollection Class](#)

[PictureAlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbumCollection.GetEnumerator Method

Returns an enumerator that iterates through the [PictureAlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<PictureAlbum> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **PictureAlbumCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[PictureAlbumCollection Class](#)

[PictureAlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[PictureAlbumCollection Class](#)



[PictureAlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbumCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of PictureAlbum objects in the PictureAlbumCollection.
	Item	Gets the PictureAlbum at the specified index in the PictureAlbumCollection.

See Also

Reference

[PictureAlbumCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureAlbumCollection.Count Property

Gets the number of [PictureAlbum](#) objects in the [PictureAlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [PictureAlbum](#) objects in this **PictureAlbumCollection**.

See Also

Reference

[PictureAlbumCollection Class](#)

[PictureAlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureAlbumCollection.Item Property

Gets the [PictureAlbum](#) at the specified index in the [PictureAlbumCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PictureAlbum this [
    int index
] { get; }
```

Property Value

A new [PictureAlbum](#) representing the picture album at the specified index in this **PictureAlbumCollection**.

Remarks

Each call returns a new [PictureAlbum](#) instance.

PictureAlbum resources such as the picture collection and children picture albums collection are cached in each **PictureAlbum** instance. Retrieving and maintaining references to multiple instances of the same picture album, and then accessing the picture collection ([PictureAlbum.Pictures](#)) or children picture albums collection ([PictureAlbum.Albums](#)) on each of those instances, creates multiple copies of the resources.

See Also

Reference

[PictureAlbumCollection Class](#)

[PictureAlbumCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureCollection Class

A collection of pictures in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class PictureCollection : IEnumerable<Picture>, IEnumerable
```

Remarks

The [PictureCollection](#) class provides access to pictures in the device's media library.

Use the [MediaLibrary.Pictures](#) property to obtain a collection of all pictures in the media library, and use the [PictureAlbum.Pictures](#) property to obtain a collection of pictures in a particular picture album.

The **PictureCollection** does not immediately instantiate instances of all pictures in the collection. Instead, individual [Picture](#) objects are created each time a picture is accessed through the collection's [Item](#) indexer. See the [Remarks](#) section of [Item](#) for the resource implications of maintaining references to multiple [Picture](#) objects.

See Also

Reference

[MediaLibrary.Pictures Property](#)

[PictureAlbum.Pictures Property](#)

[Picture Class](#)

[PictureCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








PictureCollection Members

The following tables list the members exposed by the PictureCollection type.



Public Properties

	Name	Description
	Count	Gets the number of Picture objects in the PictureCollection.
	Item	Gets the Picture at the specified index in the PictureCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the PictureCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the PictureCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[PictureCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

PictureCollection Methods


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the PictureCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the PictureCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[PictureCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[PictureCollection Class](#)

[PictureCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [PictureCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors](#), [Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[PictureCollection Class](#)

[PictureCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureCollection.GetEnumerator Method

Returns an enumerator that iterates through the [PictureCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Picture> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **PictureCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[PictureCollection Class](#)

[PictureCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[PictureCollection Class](#)



[PictureCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of Picture objects in the PictureCollection.
	Item	Gets the Picture at the specified index in the PictureCollection.

See Also

Reference

[PictureCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PictureCollection.Count Property

Gets the number of [Picture](#) objects in the [PictureCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [Picture](#) objects in this **PictureCollection**.

See Also

Reference

[PictureCollection Class](#)

[PictureCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PictureCollection.Item Property

Gets the [Picture](#) at the specified index in the [PictureCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Picture this [
    int index
] { get; }
```

Property Value

A new [Picture](#) representing the picture at the specified index in this **PictureCollection**.

Remarks

Each call returns a new [Picture](#) instance.

Picture resources such as the picture texture and thumbnail are cached in each **Picture** instance. Retrieving and maintaining references to multiple instances of the same picture, and then accessing the picture texture or thumbnail on each of those instances, creates multiple copies of the texture or thumbnail. Large numbers of picture textures can consume considerable memory, so it is not advisable to hold a large number of **Picture** objects on which [GetTexture](#) has been called.

See Also

Reference

[PictureCollection Class](#)

[PictureCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist Class

Provides access to a playlist in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Playlist : IEquatable<Playlist>
```

Remarks

The [Playlist](#) class provides information about a playlist, including the playlist's [Name](#), [Duration](#), and [Songs](#).

Obtain **Playlist** objects through the [PlaylistCollection.Item](#) indexer.

See Also

Reference

[PlaylistCollection.Item](#) Property

[Playlist](#) Members




[Microsoft.Xna.Framework.Media](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune











Playlist Members

The following tables list the members exposed by the Playlist type.



Public Properties

	Name	Description
	Duration	Gets the duration of the Playlist.
	Name	Gets the name of the Playlist.
	Songs	Gets a SongCollection that contains the songs in the playlist.

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Playlist are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
 	op_Equality	Determines whether the specified Playlist instances are equal.
 	op_Inequality	Determines whether the specified Playlist instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Playlist.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Playlist is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[Playlist Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Playlist Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Playlist are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	 op_Equality	Determines whether the specified Playlist instances are equal.
	 op_Inequality	Determines whether the specified Playlist instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Playlist.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Playlist is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Playlist Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Playlist.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.Equals Method

Determines whether two instances of [Playlist](#) are equal.

Overload List

Name	Description
Playlist.Equals (Object)	Determines whether the specified Object is equal to this Playlist .
Playlist.Equals (Playlist)	Determines whether the specified Playlist is equal to this Playlist .
Playlist.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Playlist.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [Playlist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.Equals Method (Playlist)

Determines whether the specified [Playlist](#) is equal to this **Playlist**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Playlist other  
)
```

Parameters

other

Playlist to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [Playlist](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.op_Equality Method

Determines whether the specified [Playlist](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Playlist first,  
    Playlist second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.op_Inequality Method

Determines whether the specified [Playlist](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Playlist first,  
    Playlist second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.ToString Method

Returns a [String](#) representation of the [Playlist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[Playlist Class](#)




[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist Properties

Public Properties

	Name	Description
	Duration	Gets the duration of the Playlist.
	Name	Gets the name of the Playlist.
	Songs	Gets a SongCollection that contains the songs in the playlist.

See Also

Reference

[Playlist Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Playlist.Duration Property

Gets the duration of the [Playlist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

Duration of the playlist, as a [TimeSpan](#) structure.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.Name Property

Gets the name of the [Playlist](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **Playlist**.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Playlist.Songs Property

Gets a [SongCollection](#) that contains the songs in the playlist.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SongCollection Songs { get; }
```

Property Value

[SongCollection](#) containing the songs in this playlist.

See Also

Reference

[Playlist Class](#)

[Playlist Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlaylistCollection Class

A collection of playlists in the media library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class PlaylistCollection : IEnumerable<Playlist>, IEnumerable
```

Remarks

The [PlaylistCollection](#) class provides access to playlists in the device's media library.

Use the [MediaLibrary.Playlists](#) property to obtain a collection of all playlists in the media library.

The **PlaylistCollection** does not immediately instantiate instances of all playlists in the collection. Instead, individual [Playlist](#) objects are created each time a playlist is accessed through the collection's [Item](#) indexer.

See Also

Reference

[MediaLibrary.Songs Property](#)

[Song Class](#)

[PlaylistCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








PlaylistCollection Members

The following tables list the members exposed by the PlaylistCollection type.



Public Properties

	Name	Description
	Count	Gets the number of Playlist objects in the PlaylistCollection.
	Item	Gets the Playlist at the specified index in the PlaylistCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the PlaylistCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the PlaylistCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[PlaylistCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

PlaylistCollection Methods


Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the PlaylistCollection.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the PlaylistCollection.
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[PlaylistCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PlaylistCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[PlaylistCollection Class](#)

[PlaylistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlaylistCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [PlaylistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[PlaylistCollection Class](#)

[PlaylistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlaylistCollection.GetEnumerator Method

Returns an enumerator that iterates through the [PlaylistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Playlist> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **PlaylistCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[PlaylistCollection Class](#)

[PlaylistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[PlaylistCollection Class](#)



[PlaylistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlaylistCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of Playlist objects in the PlaylistCollection.
	Item	Gets the Playlist at the specified index in the PlaylistCollection.

See Also

Reference

[PlaylistCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PlaylistCollection.Count Property

Gets the number of [Playlist](#) objects in the [PlaylistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [Playlist](#) objects in this **PlaylistCollection**.

See Also

Reference

[PlaylistCollection Class](#)

[PlaylistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PlaylistCollection.Item Property

Gets the [Playlist](#) at the specified index in the [PlaylistCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Playlist this [
    int index
] { get; }
```

Property Value

A new [Playlist](#) representing the playlist at the specified index in this **PlaylistCollection**.

Remarks

Each call returns a new [Playlist](#) instance.

The [SongCollection](#) returned by [Songs](#) is allocated and cached by each **Playlist** instance. Retrieving and maintaining references to multiple instances of the same playlist, and then accessing the song collection on each of those instances, creates multiple copies of the song collection.

See Also

Reference

[PlaylistCollection Class](#)

[PlaylistCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song Class

Provides access to a song in the song library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Song : IEquatable<Song>
```

Remarks

The [Song](#) class provides information about a song, including the song's [Name](#), [Artist](#), and [Album](#).

You can obtain a **Song** object through the [SongCollection.Item](#) indexer and the [MediaQueue.ActiveSong](#) property.

See Also

Reference

[SongCollection.Item](#) Property

[MediaQueue.ActiveSong](#) Property

[Song](#) Members











[Microsoft.Xna.Framework.Media](#) Namespace

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









Song Members

The following tables list the members exposed by the Song type.



Public Properties

Name	Description
 Album	Gets the Album on which the Song appears.
 Artist	Gets the Artist of the Song.
 Duration	Gets the duration of the Song.
 Genre	Gets the Genre of the Song.
 IsProtected	Gets a value that indicates whether the song is DRM protected content.
 IsRated	Gets a value that indicates whether the song has been rated by the user.
 Name	Gets the name of the Song.
 PlayCount	Gets the song play count.
 Rating	Gets the user's rating for the Song.
 TrackNumber	Gets the track number of the song on the song's Album .

Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	Overloaded. Determines whether two instances of Song are equal.
 GetHashCode	Gets the hash code for this instance.
 GetType	(Inherited from Object .)
 Op_Equality	Determines whether the specified Song instances are equal.
 Op_Inequality	Determines whether the specified Song instances are not equal.
 ReferenceEquals	(Inherited from Object .)
 ToString	Returns a String representation of the Song.

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before the Song is reclaimed by garbage collection.
 MemberwiseClone	(Inherited from Object .)

See Also









Reference

[Song Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Song Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	Overloaded. Determines whether two instances of Song are equal.
	GetHashCode	Gets the hash code for this instance.
	GetType	(Inherited from Object .)
	Op_Equality	Determines whether the specified Song instances are equal.
	Op_Inequality	Determines whether the specified Song instances are not equal.
	ReferenceEquals	(Inherited from Object .)
	ToString	Returns a String representation of the Song.

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before the Song is reclaimed by garbage collection.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Song Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Song.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Equals Method

Determines whether two instances of [Song](#) are equal.

Overload List

Name	Description
Song.Equals (Object)	Determines whether the specified Object is equal to this Song .
Song.Equals (Song)	Determines whether the specified Song is equal to this Song .
Song.Equals (Object, Object)	(Inherited from Object .)

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Song.Equals Method (Object)

Determines whether the specified [Object](#) is equal to this [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override bool Equals (  
    Object obj  
)
```

Parameters

obj

[Object](#) to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Equals Method (Song)

Determines whether the specified [Song](#) is equal to this **Song**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool Equals (  
    Song other  
)
```

Parameters

other

Song to compare with this instance.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Finalize Method

Releases unmanaged resources and performs other cleanup operations before the [Song](#) is reclaimed by garbage collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors](#), [Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.GetHashCode Method

Gets the hash code for this instance.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override int GetHashCode ()
```

Return Value

Hash code for this object.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.op_Equality Method

Determines whether the specified [Song](#) instances are equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Equality (  
    Song first,  
    Song second  
)
```

Parameters

first

Object to the left of the equality operator.

second

Object to the right of the equality operator.

Return Value

true if the objects are equal; **false** otherwise.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.op_Inequality Method

Determines whether the specified [Song](#) instances are not equal.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static bool op_Inequality (  
    Song first,  
    Song second  
)
```

Parameters

first

Object to the left of the inequality operator.

second

Object to the right of the inequality operator.

Return Value

true if the objects are not equal; **false** otherwise.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.ToString Method

Returns a [String](#) representation of the [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

[String](#) representation of this object.

See Also

Reference

[Song Class](#)











[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song Properties

Public Properties

	Name	Description
	Album	Gets the Album on which the Song appears.
	Artist	Gets the Artist of the Song.
	Duration	Gets the duration of the Song.
	Genre	Gets the Genre of the Song.
	IsProtected	Gets a value that indicates whether the song is DRM protected content.
	IsRated	Gets a value that indicates whether the song has been rated by the user.
	Name	Gets the name of the Song.
	PlayCount	Gets the song play count.
	Rating	Gets the user's rating for the Song.
	TrackNumber	Gets the track number of the song on the song's Album .

See Also

Reference

[Song Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Song.Album Property

Gets the [Album](#) on which the [Song](#) appears.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Album Album { get; }
```

Property Value

[Album](#) on which this **Song** appears.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Artist Property

Gets the [Artist](#) of the [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Artist Artist { get; }
```

Property Value

[Artist](#) of this **Song**.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Duration Property

Gets the duration of the [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

Duration of the song, as a [TimeSpan](#) structure.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Genre Property

Gets the [Genre](#) of the [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Genre Genre { get; }
```

Property Value

[Genre](#) of this [Song](#).

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.IsProtected Property

Gets a value that indicates whether the song is DRM protected content.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsProtected { get; }
```

Property Value

true if this [Song](#) is DRM protected; **false** otherwise.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.IsRated Property

Gets a value that indicates whether the song has been rated by the user.

Xbox 360 Specific Information

Song information is not supported. This property is always set to **false**.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsRated { get; }
```

Property Value

true if this [Song](#) has been rated by the user; **false** otherwise.

See Also

Reference

[Song.Rating Property](#)

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Name Property

Gets the name of the [Song](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The name of this **Song**.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.PlayCount Property

Gets the song play count.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PlayCount { get; }
```

Property Value

Song play count.

See Also

Reference

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.Rating Property

Gets the user's rating for the [Song](#).

Xbox 360 Specific Information

Album information is not supported. This value is always 0.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Rating { get; }
```

Property Value

User's rating for this **Song**, or 0 if the song is unrated.

Ratings range from 1 (dislike the most) to 10 (like the most).

When the user rates songs using the Zune device or Zune software, the rating is set to 8 for liked songs (shown as a heart) and to 2 or 3 for disliked songs (shown as a broken heart).

See Also

Reference

[Song.IsRated Property](#)

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Song.TrackNumber Property

Gets the track number of the song on the song's [Album](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int TrackNumber { get; }
```

Property Value

Track number of this [Song](#) on the song's [Album](#).

See Also

Reference

[Album Class](#)

[Song Class](#)

[Song Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SongCollection Class

A collection of songs in the song library.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class SongCollection : IEnumerable<Song>, IEnumerable
```

Remarks

The [SongCollection](#) class provides access to songs in the device's song library.

Use the [MediaLibrary.Songs](#) property to obtain the following collections:

- All songs in the media library.
- Songs on a particular album.
- Songs associated with a particular artist.
- Songs associated with a particular genre.

The **SongCollection** does not immediately instantiate instances of all songs in the collection. Instead, individual [Song](#) objects are created each time a user accesses a song through the collection's [Item](#) indexer. See the [Remarks](#) section of [Item](#) for the resource implications of maintaining references to multiple [Song](#) objects.

Note

All of the collections, playlists, and queues returned by methods and properties in the [Microsoft.Xna.Framework.Media](#) name space are immutable. You cannot add or remove objects from those collections or playlists. To create a custom 'playlist' of songs, games must maintain their own list of songs to play, and play those songs one at a time by calling [MediaPlayer.Play](#).

See Also

Reference

[MediaLibrary.Songs Property](#)

[Album.Songs Property](#)

[Artist.Songs Property](#)

[Genre.Songs Property](#)

[Song Class](#)

[SongCollection Members](#)



[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








SongCollection Members

The following tables list the members exposed by the SongCollection type.



Public Properties

	Name	Description
	Count	Gets the number of Song objects in the SongCollection.
	Item	Gets the Song at the specified index in the SongCollection.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetEnumerator	Returns an enumerator that iterates through the SongCollection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the SongCollection.
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also








Reference

[SongCollection Class](#)



[Microsoft.Xna.Framework.Media Namespace](#)

SongCollection Methods


Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the SongCollection.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the SongCollection.
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through the collection.

See Also

Reference

[SongCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

SongCollection.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[SongCollection Class](#)

[SongCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SongCollection.Finalize Method

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the [SongCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [FinalizeSystem.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

For more information, see [Finalize Methods and Destructors, Cleaning Up Unmanaged Resources](#), and [Overriding the Finalize Method](#).

See Also

Reference

[SongCollection Class](#)

[SongCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SongCollection.GetEnumerator Method

Returns an enumerator that iterates through the [SongCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Song> GetEnumerator ()
```

Return Value

[IEnumerator](#) for the **SongCollection**.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until **MoveNext** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. You cannot set **Current** to the first element of the collection again; you must create a new enumerator instance instead.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[SongCollection Class](#)

[SongCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

[IEnumerator](#) that iterates through the collection.

Remarks

The **foreach** statement of the C# language (**for each** in C++, **For Each** in Visual Basic) hides the complexity of enumerators. Therefore, using **foreach** instead of directly manipulating the enumerator is recommended.

Enumerators can be used to read the data in the collection, but they cannot be used to modify the underlying collection.

Initially, the enumerator is positioned before the first element in the collection. [Reset](#) also brings the enumerator back to this position. At this position, [Current](#) is undefined. You must call [MoveNext](#) to advance the enumerator to the first element of the collection before reading the value of **Current**.

Current returns the same object until either **MoveNext** or **Reset** is called. **MoveNext** sets **Current** to the next element.

If **MoveNext** passes the end of the collection, the enumerator is positioned after the last element in the collection and **MoveNext** returns false. When the enumerator is at this position, subsequent calls to **MoveNext** also return false. If the last call to **MoveNext** returned false, **Current** is undefined. To set **Current** to the first element of the collection again, you can call **Reset** followed by **MoveNext**.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

The enumerator does not have exclusive access to the collection; enumerating through a collection is intrinsically not a thread-safe procedure. To guarantee thread safety during enumeration, you can lock the collection during the entire enumeration. To allow the collection to be accessed by multiple threads for reading and writing, you must implement your own synchronization.

This method is an O(1) operation.

See Also

Reference

[SongCollection Class](#)



[SongCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SongCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of Song objects in the SongCollection.
	Item	Gets the Song at the specified index in the SongCollection.

See Also

Reference

[SongCollection Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

SongCollection.Count Property

Gets the number of [Song](#) objects in the [SongCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of [Song](#) objects in this [SongCollection](#).

See Also

Reference

[SongCollection Class](#)

[SongCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SongCollection.Item Property

Gets the [Song](#) at the specified index in the [SongCollection](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Song this [
    int index
] { get; }
```

Property Value

A new [Song](#) representing the song at the specified index in this **SongCollection**.

Remarks

Each call returns a new [Song](#) instance.

Song resources such as the album, artist, and genre are cached in each **Song** instance. Retrieving and maintaining references to multiple instances of the same song, and then accessing the album, artist, or genre on each of those instances, creates multiple copies of the album, artist, or genre. Because large numbers of album art textures can consume considerable memory, it is not advisable to hold a large number of **Song** objects on which [Song.Album.GetAlbumArt](#) has been called.

See Also

Reference

[SongCollection Class](#)

[SongCollection Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Video Class

Represents a video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class Video
```

See Also

Reference

[Video Members](#)






[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista






Video Members

The following tables list the members exposed by the Video type.



Public Properties

	Name	Description
	Duration	Gets the duration of the Video .
	FramesPerSecond	Gets the frame rate of this video.
	Height	Gets the height of this video, in pixels.
	VideoSoundtrackType	Gets the VideoSoundtrackType for this video.
	Width	Gets the width of this video, in pixels.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[Video Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Video Properties

Public Properties

	Name	Description
	Duration	Gets the duration of the Video .
	FramesPerSecond	Gets the frame rate of this video.
	Height	Gets the height of this video, in pixels.
	VideoSoundtrackType	Gets the VideoSoundtrackType for this video.
	Width	Gets the width of this video, in pixels.

See Also

Reference

[Video Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Video.Duration Property

Gets the duration of the [Video](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

Duration of the video, as a [TimeSpan](#) structure.

See Also

Reference

[Video Class](#)

[Video Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Video.FramesPerSecond Property

Gets the frame rate of this video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float FramesPerSecond { get; }
```

Property Value

The number of frames this video displays per second.

Remarks

24, 30, and 29.97 are common frame rates.

See Also

Reference

[Video Class](#)

[Video Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Video.Height Property

Gets the height of this video, in pixels.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

The height of this video, in pixels.

See Also

Reference

[Video Class](#)

[Video Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Video.VideoSoundtrackType Property

Gets the [VideoSoundtrackType](#) for this video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VideoSoundtrackType VideoSoundtrackType { get; }
```

Property Value

The type of soundtrack used in the video: dialog, music, or both.

See Also

Reference

[Video Class](#)

[Video Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

Video.Width Property

Gets the width of this video, in pixels.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

The width of this video resource, in pixels.

See Also

Reference

[Video Class](#)

[Video Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer Class

Provides methods and properties to playback, pause, resume, and stop video. [VideoPlayer](#) also exposes repeat, volume, and play position information.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class VideoPlayer : IDisposable
```

See Also

Reference

[VideoPlayer Members](#)


[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista








VideoPlayer Members

The following tables list the members exposed by the VideoPlayer type.












Public Constructors

Name	Description
 VideoPlayer	Initializes a new instance of this class.



Public Properties

Name	Description
 IsDisposed	Gets a value that indicates whether the object is disposed.
 IsLooped	Gets a value that indicates whether the player is playing video in a loop.
 IsMuted	Gets or sets the muted setting for the video player.
 PlayPosition	Gets the play position within the currently playing video.
 State	Gets the media playback state, MediaState .
 Video	Gets the Video that is currently playing.
 Volume	Gets or sets the video player volume.

Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetTexture	Retrieves a Texture2D containing the current frame of video being played.
 GetType	(Inherited from Object .)
 Pause	Pauses the currently playing video.
 Play	Plays a Video .
 ReferenceEquals	(Inherited from Object .)
 Resume	Resumes a paused video.
 Stop	Stops playing a video.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VideoPlayer Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

VideoPlayer Constructor

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VideoPlayer ()
```

See Also

Reference

[VideoPlayer Class](#)












[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetTexture	Retrieves a Texture2D containing the current frame of video being played.
	GetType	(Inherited from Object .)
	Pause	Pauses the currently playing video.
	Play	Plays a Video .
	ReferenceEquals	(Inherited from Object .)
	Resume	Resumes a paused video.
	Stop	Stops playing a video.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VideoPlayer Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

VideoPlayer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected virtual void Finalize ()
```

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.GetTexture Method

Retrieves a [Texture2D](#) containing the current frame of video being played.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Texture2D GetTexture ()
```

Return Value

The current frame of video.

Remarks

Call **GetTexture** once per frame to get the latest video data.

Use [SpriteBatch.Draw](#) to display the video on the screen, or attach the texture to an object in 3D space to display the video as part of a scene.

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Pause Method

Pauses the currently playing video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Pause ()
```

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Play Method

Plays a [Video](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Play (  
    Video video  
)
```

Parameters

video

[Video](#) to play.

Exceptions

Exception type	Condition
ArgumentNullException	<i>song</i> is null .

Remarks

After calling **Play**, call [GetTexture](#) to get the current frame of video.

See Also

Reference

[VideoPlayer Class](#)

[GetTexture](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Resume Method

Resumes a paused video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Resume ()
```

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Stop Method

Stops playing a video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Stop ()
```

See Also

Reference

[VideoPlayer Class](#)








[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	IsLooped	Gets a value that indicates whether the player is playing video in a loop.
	IsMuted	Gets or sets the muted setting for the video player.
	PlayPosition	Gets the play position within the currently playing video.
	State	Gets the media playback state, MediaState .
	Video	Gets the Video that is currently playing.
	Volume	Gets or sets the video player volume.

See Also

Reference

[VideoPlayer Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

VideoPlayer.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.IsLooped Property

Gets a value that indicates whether the player is playing video in a loop.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsLooped { get; set; }
```

Property Value

true if the player is playing video in a loop; **false** otherwise.

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.IsMuted Property

Gets or sets the muted setting for the video player.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsMuted { get; set; }
```

Property Value

true if sound is muted; **false** otherwise.

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.PlayPosition Property

Gets the play position within the currently playing video.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan PlayPosition { get; }
```

Property Value

Play position within the current song, as a [TimeSpan](#) structure.

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.State Property

Gets the media playback state, [MediaState](#).

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public MediaState State { get; }
```

Property Value

The media playback state, [MediaState](#).

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Video Property

Gets the [Video](#) that is currently playing.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Video Video { get; }
```

Property Value

The currently playing [Video](#).

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoPlayer.Volume Property

Gets or sets the video player volume.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float Volume { get; set; }
```

Property Value

Video player volume, from 0.0f (silence) to 1.0f (full volume relative to the current device volume).

Remarks

Volume adjustment is based on a decibel, not multiplicative, scale. For example, when the device volume is half of maximum (about **7** in the Zune user interface), setting **Volume** to 0.6f or less is silent or nearly so, not volume **4** as you would expect from a multiplicative adjustment.

Setting **Volume** to 0.0 subtracts 96 dB from the volume. Setting **Volume** to 1.0 subtracts 0 dB from the volume. Values in between 0.0f and 1.0f subtract dB from the volume proportionally.

See Also

Reference

[VideoPlayer Class](#)

[VideoPlayer Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VideoSoundtrackType Enumeration

Type of sounds in a video

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum VideoSoundtrackType
```

Members

Member name	Description
Dialog	This video contains only dialog.
Music	This video contains only music.
MusicAndDialog	This video contains music and dialog.

Xbox 360 Specific Information

This setting has no effect on Windows. On the Xbox 360, videos with the *Dialog* type will have their sound mixed in with the dashboard player's playback. Videos with the *Music* type will be muted if the dashboard player is playing. Videos with the *MusicAndDialog* type will override the dashboard playback.

Remarks

A video that only contains music will be muted on playback if the [MediaPlayer](#) is playing back music.

See Also

Reference

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista

VisualizationData Class

Encapsulates visualization (frequency and sample) data for the currently-playing song.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class VisualizationData
```

See Also

Reference

[VisualizationData Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



VisualizationData Members

The following tables list the members exposed by the VisualizationData type.






Public Constructors

Name	Description
 VisualizationData	Initializes a new instance of the VisualizationData class.



Public Properties

Name	Description
 Frequencies	Returns a collection of floats that contain frequency data.
 Samples	Returns a collection of floats that contain sample data.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VisualizationData Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

VisualizationData Constructor

Initializes a new instance of the [VisualizationData](#) class.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public VisualizationData ()
```

See Also

Reference

[VisualizationData Class](#)






[VisualizationData Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

VisualizationData Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[VisualizationData Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

VisualizationData Properties

Public Properties

	Name	Description
	Frequencies	Returns a collection of floats that contain frequency data.
	Samples	Returns a collection of floats that contain sample data.

See Also

Reference

[VisualizationData Class](#)

[Microsoft.Xna.Framework.Media Namespace](#)

VisualizationData.Frequencies Property

Returns a collection of floats that contain frequency data.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<float> Frequencies { get; }
```

Property Value

A collection of floats that contain frequency data.

RemarksEach value in the collection is a float value from 0.0f to 1.0f, and is the logarithmic scaled power level for that frequency band.

See Also

Reference

[VisualizationData Class](#)

[VisualizationData Members](#)

[Microsoft.Xna.Framework.Media Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

VisualizationData.Samples Property

Returns a collection of floats that contain sample data.

Sample floats are in the range -1.0f to 1.0f, which approximate the wave form of the sound. The sample data equates to the volume of the sound.

Namespace: Microsoft.Xna.Framework.Media

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public ReadOnlyCollection<float> Samples { get; }
```

Property Value

A collection of floats that contain sample data.

See Also

Reference

[VisualizationData Class](#)

[VisualizationData Members](#)





















[Microsoft.Xna.Framework.Media Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






Microsoft.Xna.Framework.Net Namespace

Contains classes that implement support for Xbox LIVE, multiplayer, and networking for XNA Framework games.

Classes

Name	Description
 AvailableNetworkSession	Describes a multiplayer session that can be joined.
 AvailableNetworkSessionCollection	Represents a collection of sessions available for joining.
 GameEndedEventArgs	Represents the arguments passed to a GameEnded event.
 GamerJoinedEventArgs	Represents the arguments passed to a GamerJoined event.
 GamerLeftEventArgs	Represents the arguments passed to a GamerLeft event.
 GameStartedEventArgs	Represents the arguments passed to a GameStarted event.
 HostChangedEventArgs	Represents the arguments passed to a HostChanged event.
 InviteAcceptedEventArgs	Represents the arguments passed to a InviteAccepted event.
 LocalNetworkGamer	Represents a local player in a network session.
 NetworkException	Thrown if there is a network communication failure.
 NetworkGamer	Represents a player in a network session.
 NetworkMachine	Represents a physical machine (such as single Xbox 360 console or Windows-based computer) that is participating in a multiplayer session. It can be used to detect when more than one NetworkGamer is playing on the same actual machine.
 NetworkNotAvailableException	Exception thrown if no network is available.
 NetworkSession	Represents a multiplayer game session.
 NetworkSessionEndedEventArgs	Represents the arguments passed to a SessionEnded event. These arguments are passed to event handlers when a session ends.
 NetworkSessionJoinException	Thrown if an error was encountered while joining a session.
 NetworkSessionProperties	Describes custom, game-specific information about a NetworkSession object.
 PacketReader	Provides common functionality for efficiently reading incoming network packets.
 PacketWriter	Provides common functionality for efficiently formatting outgoing network packets.
 QualityOfService	Describes the quality of the network connection between this machine and the host of a multiplayer session that was discovered with a matchmaking query.

Enumerations

Name	Description
 NetworkSessionEndReason	Defines the reason a session ended.
 NetworkSessionJoinError	Contains additional data about a NetworkSessionJoinException .
 NetworkSessionState	Defines the different states of a multiplayer session.
 NetworkSessionType	Defines the different types of a multiplayer session.
 SendDataOptions	Defines options for network packet transmission.

Remarks

 **Important**

Games for Windows - LIVE is not available to finished games. This functionality is not included in the redistributable version of the XNA Framework. A game that attempts to use these components without XNA Game Studio installed will result in a [GameServicesNotAvailableException](#).

See Also

Concepts

[Getting Started With Networked Games](#)

[Networking Content Catalog at XNA Creators Club Online](#)

AvailableNetworkSession Class

Describes a multiplayer session that can be joined.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class AvailableNetworkSession
```

Remarks

Use the [NetworkSession](#) class to get a list of **AvailableNetworkSession** instances. Typically, the results populate some type of user interface, displayed by the game. The user then chooses a session, and the game adds the player to the chosen session with a call to [Join](#).

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[AvailableNetworkSession Members](#)







[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






AvailableNetworkSession Members

The following tables list the members exposed by the AvailableNetworkSession type.



Public Properties

Name	Description
 CurrentGamerCount	Gets the number of gamers in the session.
 HostGamertag	Gets the gamertag of the session host.
 OpenPrivateGamerSlots	Gets the number of private player slots.
 OpenPublicGamerSlots	Gets the number of public player slots.
 QualityOfService	Gets an estimate of the quality of network service between this local machine and the remote session.
 SessionProperties	Gets any custom properties that have been attached to the session.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Find and Join a Network Session](#)






Reference

[AvailableNetworkSession Class](#)



[Microsoft.Xna.Framework.Net Namespace](#)

AvailableNetworkSession Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[AvailableNetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

AvailableNetworkSession Properties

Public Properties

	Name	Description
	CurrentGamerCount	Gets the number of gamers in the session.
	HostGamertag	Gets the gamertag of the session host.
	OpenPrivateGamerSlots	Gets the number of private player slots.
	OpenPublicGamerSlots	Gets the number of public player slots.
	QualityOfService	Gets an estimate of the quality of network service between this local machine and the remote session.
	SessionProperties	Gets any custom properties that have been attached to the session.

See Also

Reference

[AvailableNetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

AvailableNetworkSession.CurrentGamerCount Property

Gets the number of gamers in the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int CurrentGamerCount { get; }
```

Property Value

Number of gamers currently in the session.

See Also

Reference

[AvailableNetworkSession Class](#)

[AvailableNetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSession.HostGamertag Property

Gets the gamertag of the session host.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string HostGamertag { get; }
```

Property Value

Name of the gamertag.

See Also

Reference

[AvailableNetworkSession Class](#)

[AvailableNetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSession.OpenPrivateGamerSlots Property

Gets the number of private player slots. These slots are reserved for people joining by invitation or friend presence.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int OpenPrivateGamerSlots { get; }
```

Property Value

Number of reserved private slots.

See Also

Reference

[AvailableNetworkSession Class](#)

[AvailableNetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSession.OpenPublicGamerSlots Property

Gets the number of public player slots. These slots are reserved for people joining by a matchmaking service.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int OpenPublicGamerSlots { get; }
```

Property Value

Number of public slots.

See Also

Reference

[AvailableNetworkSession Class](#)

[AvailableNetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSession.QualityOfService Property

Gets an estimate of the quality of network service between this local machine and the remote session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public QualityOfService QualityOfService { get; }
```

Property Value

An estimate of the quality of network service between this local machine and the remote session. Initially the [IsAvailable](#) property will be set to false, and all the other properties of the returned [QualityOfService](#) instance will be zero, but after the probing finishes this will become true and the other properties will be filled in with some actual data.

Remarks This measurement can take some time to complete, so the data may not be available when the search result is first returned to you.

See Also

Reference

[AvailableNetworkSession Class](#)

[AvailableNetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSession.SessionProperties Property

Gets any custom properties that have been attached to the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionProperties SessionProperties { get; }
```

Property Value

Any custom properties that have been attached to the session.

See Also

Reference

[AvailableNetworkSession Class](#)

[AvailableNetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSessionCollection Class

Represents a collection of sessions available for joining.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class AvailableNetworkSessionCollection : ReadOnlyCollection<AvailableNetworkSession>, IDisposable
```

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[AvailableNetworkSessionCollection Members](#)




[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


AvailableNetworkSessionCollection Members

The following tables list the members exposed by the AvailableNetworkSessionCollection type.











Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Dispose	Immediately releases the collection.
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	Allows Finalize to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object.)

See Also

Tasks

[How To: Find and Join a Network Session](#)











Reference

[AvailableNetworkSessionCollection Class](#)



[Microsoft.Xna.Framework.Net Namespace](#)

AvailableNetworkSessionCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Dispose	Immediately releases the collection.
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	Allows Finalize to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object.)

See Also

Reference

[AvailableNetworkSessionCollection Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

AvailableNetworkSessionCollection.Dispose Method

Immediately releases the collection.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[AvailableNetworkSessionCollection Class](#)

[AvailableNetworkSessionCollection Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSessionCollection.Finalize Method

Allows **Finalize** to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[AvailableNetworkSessionCollection Class](#)




[AvailableNetworkSessionCollection Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

AvailableNetworkSessionCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[AvailableNetworkSessionCollection Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

AvailableNetworkSessionCollection.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[AvailableNetworkSessionCollection Class](#)

[AvailableNetworkSessionCollection Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameEndedEventArgs Class

Represents the arguments passed to a [GameEnded](#) event. These arguments are passed to event handlers when a game ends.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GameEndedEventArgs : EventArgs
```

See Also

Reference

[GameEndedEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GameEndedEventArgs Members

The following tables list the members exposed by the GameEndedEventArgs type.



Public Constructors

	Name	Description
	GameEndedEventArgs	Creates an instance of GameEndedEventArgs .

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameEndedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GameEndedEventArgs Constructor

Creates an instance of [GameEndedEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GameEndedEventArgs ()
```

See Also

Reference

[GameEndedEventArgs Class](#)






[GameEndedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameEndedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameEndedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerJoinedEventArgs Class

Represents the arguments passed to a [GamerJoined](#) event. These arguments are passed to event handlers when a new gamer joins a network session. This class contains the [NetworkGamer](#) instance that recently joined the network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GamerJoinedEventArgs : EventArgs
```

See Also

Reference

[GamerJoinedEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


GamerJoinedEventArgs Members

The following tables list the members exposed by the GamerJoinedEventArgs type.






Public Constructors

	Name	Description
	GamerJoinedEventArgs	Creates an instance of GamerJoinedEventArgs .



Public Properties

	Name	Description
	Gamer	Gets the gamer who just joined the session.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamerJoinedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerJoinedEventArgs Constructor

Creates an instance of [GamerJoinedEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerJoinedEventArgs (  
    NetworkGamer gamer  
)
```

Parameters

gamer

The gamer who just joined the session.

See Also

Reference

[GamerJoinedEventArgs Class](#)






[GamerJoinedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerJoinedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[GamerJoinedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerJoinedEventArgs Properties

Public Properties

	Name	Description
	Gamer	Gets the gamer who just joined the session.

See Also

Reference

[GamerJoinedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerJoinedEventArgs.Gamer Property

Gets the gamer who just joined the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkGamer Gamer { get; }
```

Property Value

The gamer who just joined the session.

See Also

Reference

[GamerJoinedEventArgs Class](#)

[GamerJoinedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerLeftEventArgs Class

Represents the arguments passed to a [GamerLeft](#) event. These arguments are passed to event handlers when a gamer leaves a network session. This class contains the [NetworkGamer](#) instance that recently left the network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GamerLeftEventArgs : EventArgs
```

See Also

Reference

[GamerLeftEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


GamerLeftEventArgs Members

The following tables list the members exposed by the GamerLeftEventArgs type.






Public Constructors

	Name	Description
	GamerLeftEventArgs	Creates an instance of GamerLeftEventArgs .



Public Properties

	Name	Description
	Gamer	Gets the gamer who just left the session.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GamerLeftEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerLeftEventArgs Constructor

Creates an instance of [GamerLeftEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerLeftEventArgs (  
    NetworkGamer gamer  
)
```

Parameters

gamer

The gamer who just left the session.

See Also

Reference

[GamerLeftEventArgs Class](#)






[GamerLeftEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GamerLeftEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[GamerLeftEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerLeftEventArgs Properties

Public Properties

	Name	Description
	Gamer	Gets the gamer who just left the session.

See Also

Reference

[GamerLeftEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GamerLeftEventArgs.Gamer Property

Gets the gamer who just left the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkGamer Gamer { get; }
```

Property Value

The gamer who just left the session.

See Also

Reference

[GamerLeftEventArgs Class](#)

[GamerLeftEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameStartedEventArgs Class

Represents the arguments passed to a [GameStarted](#) event. These arguments are passed to event handlers when a game starts.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class GameStartedEventArgs : EventArgs
```

See Also

Reference

[GameStartedEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






GameStartedEventArgs Members

The following tables list the members exposed by the GameStartedEventArgs type.



Public Constructors

Name	Description
 GameStartedEventArgs	Creates an instance of GameStartedEventArgs .

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameStartedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

GameStartedEventArgs Constructor

Creates an instance of [GameStartedEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GameStartedEventArgs ()
```

See Also

Reference

[GameStartedEventArgs Class](#)






[GameStartedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

GameStartedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GameStartedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

HostChangedEventArgs Class

Represents the arguments passed to a HostChanged event. These arguments are passed to event handlers when the host changes for a multiplayer session. This class contains the [NetworkGamer](#) instance that is the new host of the multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class HostChangedEventArgs : EventArgs
```

See Also

Reference

[HostChangedEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



HostChangedEventArgs Members

The following tables list the members exposed by the HostChangedEventArgs type.






Public Constructors

	Name	Description
	HostChangedEventArgs	Creates an instance of HostChangedEventArgs .



Public Properties

	Name	Description
	NewHost	Gets the new host of the session.
	OldHost	Gets the player who was the previous session host.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[HostChangedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

HostChangedEventArgs Constructor

Creates an instance of [HostChangedEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public HostChangedEventArgs (
    NetworkGamer oldHost,
    NetworkGamer newHost
)
```

Parameters

oldHost

The player who was the previous session host.

newHost

The player who is the new host of the multiplayer session.

See Also

Reference

[HostChangedEventArgs Class](#)






[HostChangedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HostChangedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[HostChangedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

HostChangedEventArgs Properties

Public Properties

	Name	Description
	NewHost	Gets the new host of the session.
	OldHost	Gets the player who was the previous session host.

See Also

Reference

[HostChangedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

HostChangedEventArgs.NewHost Property

Gets the new host of the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkGamer NewHost { get; }
```

Property Value

The player who is the new host of the multiplayer session.

See Also

Reference

[HostChangedEventArgs Class](#)

[HostChangedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

HostChangedEventArgs.OldHost Property

Gets the player who was the previous session host.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkGamer OldHost { get; }
```

Property Value

The player who was the previous session host.

See Also

Reference

[HostChangedEventArgs Class](#)

[HostChangedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

InviteAcceptedEventArgs Class

Represents the arguments passed to a [InviteAccepted](#) event. These arguments are passed to event handlers when a gamer accepts a game invitation. This class contains the Gamer instance that represents the gamer accepting the invitation.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class InviteAcceptedEventArgs : EventArgs
```

See Also

Reference

[InviteAcceptedEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



InviteAcceptedEventArgs Members

The following tables list the members exposed by the InviteAcceptedEventArgs type.






Public Constructors

	Name	Description
	InviteAcceptedEventArgs	Creates a new instance of InviteAcceptedEventArgs .



Public Properties

	Name	Description
	Gamer	Gets the gamer who accepted the game invitation.
	IsCurrentSession	Represents whether this invitation is for a NetworkSession that is already joined by other local gamers.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[InviteAcceptedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

InviteAcceptedEventArgs Fields

See Also

Reference

[InviteAcceptedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

InviteAcceptedEventArgs Constructor

Note

This constructor is available only when developing for Zune.

Creates a new instance of [InviteAcceptedEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public InviteAcceptedEventArgs (  
    SignedInGamer gamer  
)
```

Parameters

gamer

The player who has accepted the game invitation.

See Also

Reference

[InviteAcceptedEventArgs Class](#)






[InviteAcceptedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



PlatformsZune

InviteAcceptedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[InviteAcceptedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

InviteAcceptedEventArgs Properties

Public Properties

	Name	Description
	Gamer	Gets the gamer who accepted the game invitation.
	IsCurrentSession	Represents whether this invitation is for a NetworkSession that is already joined by other local gamers.

See Also

Reference

[InviteAcceptedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

InviteAcceptedEventArgs.Gamer Property

Gets the gamer who accepted the game invitation.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedInGamer Gamer { get; }
```

Property Value

The gamer accepting the game invitation.

See Also

Reference

[InviteAcceptedEventArgs Class](#)

[InviteAcceptedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

InviteAcceptedEventArgs.IsCurrentSession Property

Represents whether this invitation is for a `NetworkSession` that is already joined by other local gamers.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsCurrentSession { get; }
```

Property Value

true if this invitation is for a pre-existing session with other local players on the current machine; **false**, otherwise.

Remarks

If `IsCurrentSession` is **false**, indicating that the invitation is from a session that hasn't already been joined, call `NetworkSession.JoinInvited` to allow the gamer to join the remote session. If `IsCurrentSession` is **true**, indicating that another local gamer has already joined the session, call `NetworkSession.AddLocalGamer` to add additional gamers to the session.

See Also

Reference

[InviteAcceptedEventArgs Class](#)

[InviteAcceptedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer Class

Represents a local player in a network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class LocalNetworkGamer : NetworkGamer
```

See Also

Reference

[SignedInGamer](#)

[LocalNetworkGamer Members](#)




















[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune












LocalNetworkGamer Members

The following tables list the members exposed by the LocalNetworkGamer type.



Public Properties

Name	Description
 Gamertag	(Inherited from Gamer .)
 HasLeftSession	(Inherited from NetworkGamer .)
 HasVoice	(Inherited from NetworkGamer .)
 Id	(Inherited from NetworkGamer .)
 IsDataAvailable	Determines if there is incoming packet data for this player.
 IsDisposed	(Inherited from Gamer .)
 IsGuest	(Inherited from NetworkGamer .)
 IsHost	(Inherited from NetworkGamer .)
 IsLocal	(Inherited from NetworkGamer .)
 IsMutedByLocalUser	(Inherited from NetworkGamer .)
 IsPrivateSlot	(Inherited from NetworkGamer .)
 IsReady	(Inherited from NetworkGamer .)
 IsTalking	(Inherited from NetworkGamer .)
 Machine	(Inherited from NetworkGamer .)
 RoundtripTime	(Inherited from NetworkGamer .)
 Session	(Inherited from NetworkGamer .)
 SignedInGamer	Gets the SignedInGamer instance for this SignedInGamer object.
 SignedInGamers	(Inherited from Gamer .)
 Tag	(Inherited from Gamer .)

Public Methods

Name	Description
 BeginGetProfile	(Inherited from Gamer .)
 EnableSendVoice	Specifies whether voice data should be sent to, or received from, the specified remote gamer.
 EndGetProfile	(Inherited from Gamer .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetProfile	(Inherited from Gamer .)
 GetType	(Inherited from Object .)
 ReceiveData	Overloaded. Reads the next incoming packet.
 ReferenceEquals	(Inherited from Object .)
 SendData	Overloaded. Sends data to a specified set of gamers in a network session.
 SendPartyInvites	Sends game invitations to all party members that are not in the current game session.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also












Reference

[LocalNetworkGamer Class](#)



[Microsoft.Xna.Framework.Net Namespace](#)

LocalNetworkGamer Methods

Public Methods

	Name	Description
	BeginGetProfile	(Inherited from Gamer .)
	EnableSendVoice	Specifies whether voice data should be sent to, or received from, the specified remote gamer.
	EndGetProfile	(Inherited from Gamer .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetProfile	(Inherited from Gamer .)
	GetType	(Inherited from Object .)
	ReceiveData	Overloaded. Reads the next incoming packet.
	ReferenceEquals	(Inherited from Object .)
	SendData	Overloaded. Sends data to a specified set of gamers in a network session.
	SendPartyInvites	Sends game invitations to all party members that are not in the current game session.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[LocalNetworkGamer Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

LocalNetworkGamer.EnableSendVoice Method

Specifies whether voice data should be sent to, or received from, the specified remote gamer.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void EnableSendVoice (
    NetworkGamer remoteGamer,
    bool enable
)
```

Parameters

remoteGamer

A remote gamer in the session.

enable

true if voice is enabled; **false** otherwise.

Exceptions

Exception type	Condition
ArgumentException	This NetworkGamer belongs to a different network session.
ArgumentNullException	<i>remoteGamer</i> is null .
ObjectDisposedException	This NetworkGamer is no longer valid. The gamer may have left the session.

Remarks

Call this method for local gamer instances only.

By default, voice is enabled for all gamers. However, you can selectively turn it off to enable features such as team chat. Also, you can dynamically turn it on or off, depending on the player's location in the game world, to achieve a proximity voice effect.

See Also

Reference

[GamerPrivileges.AllowCommunication Property](#)

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.ReceiveData Method

Reads the next incoming packet.

Overload List

Name	Description
LocalNetworkGamer.ReceiveData (Byte[], Int32, NetworkGamer)	Reads the next incoming packet using the specified values.
LocalNetworkGamer.ReceiveData (Byte[], NetworkGamer)	Reads the next incoming packet using the specified values.
LocalNetworkGamer.ReceiveData (PacketReader, NetworkGamer)	Reads the next incoming packet and copies the packet data into the specified reader object.

Example

Be sure to call **ReceiveData** for all incoming data for all local gamers, even if the data is to be discarded. Any time data is sent to a local network gamer, the incoming data queue will grow until the data is read. If the data is not read using **ReceiveData** the performance of the application will degrade as the data queue grows in size.

C#

```
foreach (LocalNetworkGamer gamer in session.LocalGamers)
{
    // Keep reading while packets are available.
    while (gamer.IsDataAvailable)
    {
        NetworkGamer sender;

        // Read a single packet.
        gamer.ReceiveData(packetReader, out sender);
        // Discard the data, or use packetReader.Read* to process the data.
    }
    gamer.SendData(packetWriter, SendDataOptions.None);
}
```

See Also

Tasks

[How To: Receive Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

LocalNetworkGamer.ReceiveData Method (Byte[], Int32, NetworkGamer)

Reads the next incoming packet using the specified values.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ReceiveData (
    byte[] data,
    int offset,
    out NetworkGamer sender
)
```

Parameters

data

Storage for the network packet data.

offset

Offset, in bytes, to begin reading data.

sender

[[OutAttribute](#)] Gamer who sent this packet.

Return Value

Number of bytes read from the network packet. Returns 0 if no packet is available.

Exceptions

Exception type	Condition
ArgumentException	The specified array is too small to receive the incoming network packet.
ArgumentNullException	<i>data</i> is null .
ArgumentOutOfRangeException	<i>offset</i> cannot be larger than the amount of data available for storage.

Example

Be sure to call **ReceiveData** for all incoming data for all local gamers, even if the data is to be discarded. Any time data is sent to a local network gamer, the incoming data queue will grow until the data is read. If the data is not read using **ReceiveData** the performance of the application will degrade as the data queue grows in size.

C#

```
foreach (LocalNetworkGamer gamer in session.LocalGamers)
{
    // Keep reading while packets are available.
    while (gamer.IsDataAvailable)
    {
        NetworkGamer sender;

        // Read a single packet.
        gamer.ReceiveData(packetReader, out sender);
        // Discard the data, or use packetReader.Read* to process the data.
    }
    gamer.SendData(packetWriter, SendDataOptions.None);
}
```

Remarks

ReceiveData copies the packet data into the array specified by *data*, returning the number of bytes read. If several independent packets are available, only one is returned for each **ReceiveData** call.

See Also

Tasks

[How To: Receive Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.ReceiveData Method (Byte[], NetworkGamer)

Reads the next incoming packet using the specified values.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ReceiveData (
    byte[] data,
    out NetworkGamer sender
)
```

Parameters

data

Storage for the network packet data.

sender

[[OutAttribute](#)] Gamer who sent this packet.

Return Value

Number of bytes read from the network packet. Returns 0 if no packet is available.

Exceptions

Exception type	Condition
ArgumentException	The specified array is too small to receive the incoming network packet.
ArgumentNullException	<i>data</i> is null .

Example

Be sure to call **ReceiveData** for all incoming data for all local gamers, even if the data is to be discarded. Any time data is sent to a local network gamer, the incoming data queue will grow until the data is read. If the data is not read using **ReceiveData** the performance of the application will degrade as the data queue grows in size.

C#

```
foreach (LocalNetworkGamer gamer in session.LocalGamers)
{
    // Keep reading while packets are available.
    while (gamer.IsDataAvailable)
    {
        NetworkGamer sender;

        // Read a single packet.
        gamer.ReceiveData(packetReader, out sender);
        // Discard the data, or use packetReader.Read* to process the data.
    }
    gamer.SendData(packetWriter, SendDataOptions.None);
}
```

Remarks

ReceiveData copies the packet data into the array specified by *data*, returning the number of bytes read. If several independent packets are available, only one is returned for each **ReceiveData** call.

See Also

Tasks

[How To: Receive Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.ReceiveData Method (PacketReader, NetworkGamer)

Reads the next incoming packet and copies the packet data into the specified reader object.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int ReceiveData (
    PacketReader data,
    out NetworkGamer sender
)
```

Parameters

data

Network packet data.

sender

[[OutAttribute](#)] Gamer who sent this packet.

Return Value

Number of bytes read from the network packet. Returns 0 if no packet is available.

Exceptions

Exception type	Condition
ArgumentException	The specified array is too small to receive the incoming network packet.
ArgumentNullException	<i>data</i> is null .

Example

Be sure to call **ReceiveData** for all incoming data for all local gamers, even if the data is to be discarded. Any time data is sent to a local network gamer, the incoming data queue will grow until the data is read. If the data is not read using **ReceiveData** the performance of the application will degrade as the data queue grows in size.

C#

```
foreach (LocalNetworkGamer gamer in session.LocalGamers)
{
    // Keep reading while packets are available.
    while (gamer.IsDataAvailable)
    {
        NetworkGamer sender;

        // Read a single packet.
        gamer.ReceiveData(packetReader, out sender);
        // Discard the data, or use packetReader.Read* to process the data.
    }
    gamer.SendData(packetWriter, SendDataOptions.None);
}
```

RemarksIf several independent packets are available, only one will be returned on each call to **ReceiveData**.

See Also

Tasks

[How To: Receive Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendData Method

Sends data to a specified set of gamers in a network session.

Overload List

Name	Description
LocalNetworkGamer.SendData (Byte[], Int32, Int32, SendDataOptions)	Sends a selected portion of a byte array to all gamers in a session.
LocalNetworkGamer.SendData (Byte[], Int32, Int32, SendDataOptions, NetworkGamer)	Sends a selected portion of a byte array to the specified gamer.
LocalNetworkGamer.SendData (Byte[], SendDataOptions)	Sends a byte array to all gamers in a session.
LocalNetworkGamer.SendData (Byte[], SendDataOptions, NetworkGamer)	Sends a byte array to the specified gamer.
LocalNetworkGamer.SendData (PacketWriter, SendDataOptions)	Sends a network packet to all gamers in a session.
LocalNetworkGamer.SendData (PacketWriter, SendDataOptions, NetworkGamer)	Sends a network packet to the specified gamer in a session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

LocalNetworkGamer.SendData Method (Byte[], Int32, Int32, SendDataOptions)

Sends a selected portion of a byte array to all gamers in a session. This includes the gamer sending the packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendData (
    byte[] data,
    int offset,
    int count,
    SendDataOptions options
)
```

Parameters

data

Byte array containing session data.

offset

Offset, in bytes, to the start of the data.

count

Amount, in bytes, of data sent.

options

Enumeration containing data send options.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
ArgumentOutOfRangeException	<ul style="list-style-type: none"> <i>offset</i> must be greater than zero, and no larger than the amount of data available. <i>count</i> must be greater than zero, and <i>offset</i> + <i>count</i> must be no larger than the amount of data available.
ObjectDisposedException	The NetworkGamer sending this message is no longer valid. The gamer may have left the session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendData Method (Byte[], Int32, Int32, SendDataOptions, NetworkGamer)

Sends a selected portion of a byte array to the specified gamer.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendData (
    byte[] data,
    int offset,
    int count,
    SendDataOptions options,
    NetworkGamer recipient
)
```

Parameters

data

Byte array containing session data.

offset

Offset, in bytes, to the start of the data.

count

Amount, in bytes, of data sent.

options

Enumeration containing data send options.

recipient

Gamer to receive the data packet.

Exceptions

Exception type	Condition
ArgumentException	The NetworkGamer specified as the recipient does not belong to this network session.
ArgumentNullException	<i>data</i> or <i>recipient</i> is null .
ArgumentOutOfRangeException	<ul style="list-style-type: none"> <i>offset</i> must be greater than zero, and no larger than the amount of data available. <i>count</i> must be greater than zero, and <i>offset</i> + <i>count</i> must be no larger than the amount of data available.
ObjectDisposedException	The NetworkGamer specified as either the sender or the recipient is no longer valid. The gamer may have left the session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendData Method (Byte[], SendDataOptions)

Sends a byte array to all gamers in a session. This includes the gamer sending the packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendData (  
    byte[] data,  
    SendDataOptions options  
)
```

Parameters

data

Byte array containing session data.

options

Enumeration containing data send options.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
ObjectDisposedException	The NetworkGamer sending this message is no longer valid. The gamer may have left the session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendData Method (Byte[], SendDataOptions, NetworkGamer)

Sends a byte array to the specified gamer.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendData (  
    byte[] data,  
    SendDataOptions options,  
    NetworkGamer recipient  
)
```

Parameters

data

Byte array containing session data.

options

Enumeration containing data send options.

recipient

Gamer to receive the data packet.

Exceptions

Exception type	Condition
ArgumentException	The NetworkGamer specified as the recipient does not belong to this network session.
ArgumentNullException	<i>data</i> or <i>recipient</i> is null .
ObjectDisposedException	The NetworkGamer specified as either the sender or the recipient is no longer valid. The gamer may have left the session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendData Method (PacketWriter, SendDataOptions)

Sends a network packet to all gamers in a session. This includes the gamer sending the packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendData (  
    PacketWriter data,  
    SendDataOptions options  
)
```

Parameters

data

Byte array containing session data.

options

Enumeration containing data send options.

Exceptions

Exception type	Condition
ArgumentNullException	<i>data</i> is null .
ObjectDisposedException	The NetworkGamer sending this message is no longer valid. The gamer may have left the session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendData Method (PacketWriter, SendDataOptions, NetworkGamer)

Sends a network packet to the specified gamer in a session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendData (  
    PacketWriter data,  
    SendDataOptions options,  
    NetworkGamer recipient  
)
```

Parameters

data

Byte array containing session data.

options

Enumeration containing data send options.

recipient

Gamer receiving the data.

Exceptions

Exception type	Condition
ArgumentException	The NetworkGamer specified as the recipient does not belong to this network session.
ArgumentNullException	<i>data</i> or <i>recipient</i> is null .
ObjectDisposedException	The NetworkGamer specified as either the sender or the recipient is no longer valid. The gamer may have left the session.

See Also

Tasks

[How To: Send Data](#)

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SendPartyInvites Method

Sends game invitations to all party members that are not in the current game session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void SendPartyInvites ()
```

Remarks

This method has no effect when the player is not currently a member of a LIVE party, or on platforms that do not support LIVE parties.

See Also

Tasks

[How To: Add LIVE Party Support](#)

[How To: Add Support for Game Invitations](#)

Reference

[LocalNetworkGamer Class](#)




















[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer Properties

Public Properties

	Name	Description
	Gamertag	(Inherited from Gamer .)
	HasLeftSession	(Inherited from NetworkGamer .)
	HasVoice	(Inherited from NetworkGamer .)
	Id	(Inherited from NetworkGamer .)
	IsDataAvailable	Determines if there is incoming packet data for this player.
	IsDisposed	(Inherited from Gamer .)
	IsGuest	(Inherited from NetworkGamer .)
	IsHost	(Inherited from NetworkGamer .)
	IsLocal	(Inherited from NetworkGamer .)
	IsMutedByLocalUser	(Inherited from NetworkGamer .)
	IsPrivateSlot	(Inherited from NetworkGamer .)
	IsReady	(Inherited from NetworkGamer .)
	IsTalking	(Inherited from NetworkGamer .)
	Machine	(Inherited from NetworkGamer .)
	RoundtripTime	(Inherited from NetworkGamer .)
	Session	(Inherited from NetworkGamer .)
	SignedInGamer	Gets the SignedInGamer instance for this SignedInGamer object.
	SignedInGamers	(Inherited from Gamer .)
	Tag	(Inherited from Gamer .)

See Also

Reference

[LocalNetworkGamer Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

LocalNetworkGamer.IsDataAvailable Property

Determines if there is incoming packet data for this player.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDataAvailable { get; }
```

Property Value

true if there are one or more packets waiting; otherwise **false**.

See Also

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

LocalNetworkGamer.SignedInGamer Property

Gets the [SignedInGamer](#) instance for this **SignedInGamer** object.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public SignedInGamer SignedInGamer { get; }
```

Property Value

The related [SignedInGamer](#) instance.

See Also

Reference

[LocalNetworkGamer Class](#)

[LocalNetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkException Class

Thrown if there is a network communication failure.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public class NetworkException : Exception
```

See Also

Reference

[NetworkException Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








NetworkException Members

The following tables list the members exposed by the NetworkException type.


Public Constructors

Name	Description
 NetworkException	Overloaded. Creates an instance of NetworkException .







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkException Constructor

Creates an instance of [NetworkException](#).

Overload List

Name	Description
NetworkException ()	Creates an empty instance of NetworkException .
NetworkException (SerializationInfo, StreamingContext)	Initializes a new instance of NetworkException with the specified streaming context.
NetworkException (String)	Initializes a new instance of NetworkException with the specified error message.
NetworkException (String, Exception)	Initializes a new instance of NetworkException with the specified error message and the inner exception.

See Also

Reference

[NetworkException Class](#)

[NetworkException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkException Constructor ()

Creates an empty instance of [NetworkException](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkException ()
```

See Also

Reference

[NetworkException Class](#)

[NetworkException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [NetworkException](#) with the specified streaming context.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected NetworkException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Describes the network data being sent or received when the exception occurred.

context

Describes the stream where the exception occurred.

See Also

Reference

[NetworkException Class](#)

[NetworkException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NetworkException Constructor (String)

Initializes a new instance of [NetworkException](#) with the specified error message.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[NetworkException Class](#)

[NetworkException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkException Constructor (String, Exception)

Initializes a new instance of [NetworkException](#) with the specified error message and the inner exception.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The inner exception related to this exception.

See Also

Reference

[NetworkException Class](#)







[NetworkException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[NetworkException Class](#)


[Microsoft.Xna.Framework.Net Namespace](#)

NetworkException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[NetworkException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkGamer Class

Represents a player in a network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class NetworkGamer : Gamer
```

RemarksThe [NetworkSession](#) class maintains a list of players in a network session. Some of the players may also be local players, which are represented [LocalNetworkGamer](#).

See Also

Reference

[NetworkGamer Members](#)


















[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune








NetworkGamer Members

The following tables list the members exposed by the NetworkGamer type.



Public Properties

Name	Description
 Gamertag	(Inherited from Gamer .)
 HasLeftSession	Indicates whether this gamer has left the session.
 HasVoice	Determines if the player has a voice headset.
 Id	Gets a unique identifier that can be used to refer to this gamer in network packets.
 IsDisposed	(Inherited from Gamer .)
 IsGuest	Determines whether this gamer is logged in as a guest profile.
 IsHost	Determines if the player is the host of the multiplayer session.
 IsLocal	Determines if the player is playing on a local machine.
 IsMutedByLocalUser	Determines if the player is muted by one or more local users.
 IsPrivateSlot	Determines if the player occupies a reserved private session slot.
 IsReady	Determines whether the gamer is ready to leave the lobby screen and begin gameplay.
 IsTalking	Determines whether the gamer is currently sending voice data.
 Machine	Gets an object representing the physical gaming machine this NetworkGamer is playing on.
 RoundtripTime	Gets an estimate of the network latency involved in sending a packet round trip from the local machine to this gamer and back again.
 Session	Gets the multiplayer session of the gamer.
 SignedInGamers	(Inherited from Gamer .)
 Tag	(Inherited from Gamer .)

Public Methods

Name	Description
 BeginGetProfile	(Inherited from Gamer .)
 EndGetProfile	(Inherited from Gamer .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetProfile	(Inherited from Gamer .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also








Reference

[NetworkGamer Class](#)



[Microsoft.Xna.Framework.Net Namespace](#)

NetworkGamer Methods

Public Methods

	Name	Description
	BeginGetProfile	(Inherited from Gamer.)
	EndGetProfile	(Inherited from Gamer.)
	Equals	(Inherited from Object.)
	GetHashCode	(Inherited from Object.)
	GetProfile	(Inherited from Gamer.)
	GetType	(Inherited from Object.)
	ReferenceEquals	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also


















Reference

[NetworkGamer Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkGamer Properties

Public Properties

	Name	Description
	Gamertag	(Inherited from Gamer .)
	HasLeftSession	Indicates whether this gamer has left the session.
	HasVoice	Determines if the player has a voice headset.
	Id	Gets a unique identifier that can be used to refer to this gamer in network packets.
	IsDisposed	(Inherited from Gamer .)
	IsGuest	Determines whether this gamer is logged in as a guest profile.
	IsHost	Determines if the player is the host of the multiplayer session.
	IsLocal	Determines if the player is playing on a local machine.
	IsMutedByLocalUser	Determines if the player is muted by one or more local users.
	IsPrivateSlot	Determines if the player occupies a reserved private session slot.
	IsReady	Determines whether the gamer is ready to leave the lobby screen and begin gameplay.
	IsTalking	Determines whether the gamer is currently sending voice data.
	Machine	Gets an object representing the physical gaming machine this NetworkGamer is playing on.
	RoundtripTime	Gets an estimate of the network latency involved in sending a packet round trip from the local machine to this gamer and back again.
	Session	Gets the multiplayer session of the gamer.
	SignedInGamers	(Inherited from Gamer .)
	Tag	(Inherited from Gamer .)

See Also

Reference

[NetworkGamer Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkGamer.HasLeftSession Property

Indicates whether this gamer has left the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasLeftSession { get; }
```

Property Value

true if the player has left the session; **false**, otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.HasVoice Property

Determines if the player has a voice headset.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool HasVoice { get; }
```

Property Value

true if a headset is present; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.Id Property

Gets a unique identifier that can be used to refer to this gamer in network packets.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public byte Id { get; }
```

Property Value

A unique identifier that can be used to refer to this gamer in network packets.

Remarks This value is synchronized across all the machines in the session, so it can be used in situations where gamer A wants to send a packet to B, and the contents of this packet must refer to gamer C. Using the identifier is preferable to sending the index of the gamer within the session, because indices could change if gamers join or leave in mid session, but this id will always remain constant. To look up a gamer once you know their id, use [FindGamerById](#).

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsGuest Property

Determines whether this gamer is logged in as a guest profile.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsGuest { get; }
```

Property Value

true if this gamer is logged in as a guest profile; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsHost Property

Determines if the player is the host of the multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsHost { get; }
```

Property Value

true if the player is the host; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsLocal Property

Determines if the player is playing on a local machine.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsLocal { get; }
```

Property Value

true if the player is local; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsMutedByLocalUser Property

Determines if the player is muted by one or more local users.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsMutedByLocalUser { get; }
```

Property Value

true if the player is muted by one or more players in the session; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsPrivateSlot Property

Determines if the player occupies a reserved private session slot.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsPrivateSlot { get; }
```

Property Value

true if the player occupies a reserved private session slot; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsReady Property

Determines whether the gamer is ready to leave the lobby screen and begin gameplay.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsReady { get; set; }
```

Property Value

true if the gamer is ready to start playing; **false** otherwise.

Exceptions

Exception type	Condition
InvalidOperationException	<ul style="list-style-type: none"> This operation is only valid when the session state is NetworkSessionState.Lobby. This method cannot be called on remote gamer instances. It is only valid when NetworkGamer.IsLocal is true. If you are looping over the contents of the NetworkSession.AllGamers collection, consider using NetworkSession.LocalGamers instead.
ObjectDisposedException	This NetworkGamer is no longer valid. The gamer may have left the session.

Remarks

The host of a multiplayer session can use the [IsEveryoneReady](#) property to detect when all players have marked themselves ready, and then call [StartGame](#) to kick off the game.

See Also

Tasks

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.IsTalking Property

Determines whether the gamer is currently sending voice data.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsTalking { get; }
```

Property Value

true if the gamer is sending voice data; **false** otherwise.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.Machine Property

Gets an object representing the physical gaming machine this [NetworkGamer](#) is playing on.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkMachine Machine { get; set; }
```

Property Value

An object representing the physical gaming machine this [NetworkGamer](#) is playing on.

Remarks If there are several gamers all playing on the same gaming machine, they will all share a single value for the

Machine property.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.RoundtripTime Property

Gets an estimate of the network latency involved in sending a packet round trip from the local machine to this gamer and back again.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan RoundtripTime { get; }
```

Property Value

An estimate of the network latency involved in sending a packet round trip from the local machine to this gamer and back again.

Remarks This value does not include latency introduced via the [NetworkSession.SimulatedLatency](#) property.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkGamer.Session Property

Gets the multiplayer session of the gamer.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSession Session { get; }
```

Property Value

Session that contains the gamer.

See Also

Reference

[NetworkGamer Class](#)

[NetworkGamer Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkMachine Class

Represents a physical machine (such as single Xbox 360 console or Windows-based computer) that is participating in a multiplayer session. It can be used to detect when more than one [NetworkGamer](#) is playing on the same actual machine.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class NetworkMachine
```

See Also

Reference

[NetworkMachine Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







NetworkMachine Members

The following tables list the members exposed by the NetworkMachine type.



Public Properties

	Name	Description
	Gamers	Gets a collection of all the gamers who are playing on this machine.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	RemoveFromSession	Forcibly removes this machine from the session.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[NetworkMachine Class](#)



[Microsoft.Xna.Framework.Net Namespace](#)

NetworkMachine Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	RemoveFromSession	Forcibly removes this machine from the session.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkMachine Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkMachine.RemoveFromSession Method

Forcibly removes this machine from the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void RemoveFromSession ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	There are no gamers on this machine, or the gamer is no longer valid.
InvalidOperationException	<ul style="list-style-type: none">The local NetworkMachine cannot be removed from the session.This method can only be called by the session host.

See Also

Reference

[NetworkMachine Class](#)


[NetworkMachine Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkMachine Properties

Public Properties

	Name	Description
	Gamers	Gets a collection of all the gamers who are playing on this machine.

See Also

Reference

[NetworkMachine Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkMachine.Gamers Property

Gets a collection of all the gamers who are playing on this machine.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerCollection<NetworkGamer> Gamers { get; }
```

Property Value

A collection of all the gamers who are playing on this machine.

See Also

Reference

[NetworkMachine Class](#)

[NetworkMachine Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkNotAvailableException Class

Exception thrown if no network is available. This can occur on a gaming machine if the network cable is unplugged, or on Zune if wireless is disabled.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public class NetworkNotAvailableException : NetworkException
```

See Also

Reference

[NetworkNotAvailableException Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








NetworkNotAvailableException Members

The following tables list the members exposed by the NetworkNotAvailableException type.


Public Constructors

Name	Description
 NetworkNotAvailableException	Overloaded. Initializes a new instance of the NetworkNotAvailableException class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkNotAvailableException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkNotAvailableException Constructor

Initializes a new instance of the [NetworkNotAvailableException](#) class.

Overload List

Name	Description
NetworkNotAvailableException ()	Constructs an empty exception.
NetworkNotAvailableException (SerializationInfo, StreamingContext)	Constructs an exception from the specified streaming context.
NetworkNotAvailableException (String)	Constructs an exception with the specified error message.
NetworkNotAvailableException (String, Exception)	Constructs an exception with the specified error message and inner exception.

See Also

Reference

[NetworkNotAvailableException Class](#)

[NetworkNotAvailableException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkNotAvailableException Constructor ()

Constructs an empty exception.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkNotAvailableException ()
```

See Also

Reference

[NetworkNotAvailableException Class](#)

[NetworkNotAvailableException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkNotAvailableException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Constructs an exception from the specified streaming context.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected NetworkNotAvailableException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Serialization information.

context

Streaming context.

See Also

Reference

[NetworkNotAvailableException Class](#)

[NetworkNotAvailableException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NetworkNotAvailableException Constructor (String)

Constructs an exception with the specified error message.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkNotAvailableException (  
    string message  
)
```

Parameters

message

Error message to associate with this exception.

See Also

Reference

[NetworkNotAvailableException Class](#)

[NetworkNotAvailableException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkNotAvailableException Constructor (String, Exception)

Constructs an exception with the specified error message and inner exception.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkNotAvailableException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

Error message to associate with this exception.

innerException

Inner exception.

See Also

Reference

[NetworkNotAvailableException Class](#)







[NetworkNotAvailableException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkNotAvailableException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[NetworkNotAvailableException Class](#)


[Microsoft.Xna.Framework.Net Namespace](#)

NetworkNotAvailableException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[NetworkNotAvailableException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession Class

Represents a multiplayer game session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class NetworkSession : IDisposable
```

Remarks

An XNA Framework game can only initiate a single multiplayer session at a time. To start a multiplayer session, make a call to [Create](#) or use [Find](#) or [BeginFind](#) to search for and join an existing network session.

See Also

Tasks

[How To: Create a Network Session](#)

[How To: Find and Join a Network Session](#)

Concepts

[Network Session Management](#)

Reference

[NetworkSession Members](#)



[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




















NetworkSession Members

The following tables list the members exposed by the NetworkSession type.


Public Fields




































Name	Description
 MaxPreviousGamers	Represents the maximum number of gamers that can be held in the PreviousGamers property.
 MaxSupportedGamers	Maximum number of gamers supported in a session.

Public Properties



Name	Description
 AllGamers	Gets the collection of gamers currently in the session.
 AllowHostMigration	Gets or sets whether host migration is allowed. This can be read by any machine in the session, but can only be changed by the host. The default value is false indicating that host migration is disabled.
 AllowJoinInProgress	Gets or sets whether join-in-progress is allowed. If the host enables this setting, new machines will be able to join at any time. The default value is false, indicating that join-in-progress is disabled. AllowJoinInProgress can be read by any machine in the session, but can only be changed by the host.
 BytesPerSecondReceived	Gets a performance counter recording the amount of data being received from the network.
 BytesPerSecondSent	Gets a performance counter recording the amount of data being sent over the network.
 Host	Gets the current host of the multiplayer session.
 IsDisposed	Gets a value that indicates whether the object is disposed.
 IsEveryoneReady	Determines whether all gamers are ready to enter the session.
 IsHost	Determines whether this machine is the session host.
 LocalGamers	Get the collection of local gamers for a multiplayer session.
 MaxGamers	Gets or sets the maximum number of players able to join this multiplayer session.
 PreviousGamers	A collection of previous gamers in the network session.
 PrivateGamerSlots	Gets or sets the number of private slots reserved for gamers who join using an invitation.
 RemoteGamers	Gets the collection of remote gamers for a multiplayer session.
 SessionProperties	Gets any custom properties that have been attached to the session.
 SessionState	Gets the current state of a multiplayer session.
 SessionType	Gets the current multiplayer session type.
 SimulatedLatency	Gets or sets the amount of simulated network latency.
 SimulatedPacketLoss	Gets or sets the amount of simulated packet loss.

Public Methods









Name	Description
 AddLocalGamer	Adds the specified local gamer profile to the network session.

 	BeginCreate	Overloaded. Starts hosting a new multiplayer session.
 	BeginFind	Overloaded. Starts a matchmaking query to search for available multiplayer sessions.
 	BeginJoin	Starts a join operation for the specified multiplayer session.
 	BeginJoinInvited	Overloaded. Starts joining an existing network session in response to an InviteAccepted notification event. Call EndJoinInvited to access the asynchronous method results.
 	Create	Overloaded. Hosts a new multiplayer session.
	Dispose	Ends the current multiplayer session.
 	EndCreate	Gets the result from a BeginCreate asynchronous call.
 	EndFind	Gets the result from a BeginFind asynchronous call.
	EndGame	Changes the session state from NetworkSessionState.Playing to NetworkSessionState.Lobby .
 	EndJoin	Gets the result from a BeginJoin asynchronous call.
 	EndJoinInvited	Gets the result from a BeginJoinInvited asynchronous call.
	Equals	(Inherited from Object .)
 	Find	Overloaded. Issues a matchmaking query, searching for available multiplayer sessions.
	FindGamerById	Looks up the network gamer with the specified ID.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
 	Join	Joins an existing multiplayer session.
 	JoinInvited	Overloaded. Joins the specified local gamers, along with the local machine, to an existing network session in response to an InviteAccepted notification event from a non-local session.
	ReferenceEquals	(Inherited from Object .)
	ResetReady	Resets the IsReady property of all session gamers.
	StartGame	Changes the session state from NetworkSessionState.Lobby to NetworkSessionState.Playing .
	ToString	(Inherited from Object .)
	Update	Updates the state of the multiplayer session.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

Public Events

	Name	Description
	GameEnded	Occurs when the game moves from gameplay to the lobby.
	GamerJoined	Occurs when a new player joins a multiplayer session.
	GamerLeft	Occurs when a player leaves the multiplayer session.
	GameStarted	Occurs when the game moves from the lobby into actual gameplay.
	HostChanged	Occurs when the session host has changed.
 	InviteAccepted	Event that occurs when a user has accepted an invitation to join a network session.
	SessionEnded	Occurs when the multiplayer session ends.

See Also



Reference

[NetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession Fields

Public Fields

	Name	Description
	MaxPreviousGamers	Represents the maximum number of gamers that can be held in the PreviousGamers property.
	MaxSupportedGamers	Maximum number of gamers supported in a session.

See Also

Reference

[NetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.MaxPreviousGamers Field

Represents the maximum number of gamers that can be held in the [PreviousGamers](#) property.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxPreviousGamers
```

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.MaxSupportedGamers Field

Maximum number of gamers supported in a session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public const int MaxSupportedGamers
```

See Also

Reference

[NetworkSession Class](#)













[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession Methods

Public Methods

	Name	Description
	AddLocalGamer	Adds the specified local gamer profile to the network session.
	BeginCreate	Overloaded. Starts hosting a new multiplayer session.
	BeginFind	Overloaded. Starts a matchmaking query to search for available multiplayer sessions.
	BeginJoin	Starts a join operation for the specified multiplayer session.
	BeginJoinInvited	Overloaded. Starts joining an existing network session in response to an InviteAccepted notification event. Call EndJoinInvited to access the asynchronous method results.
	Create	Overloaded. Hosts a new multiplayer session.
	Dispose	Ends the current multiplayer session.
	EndCreate	Gets the result from a BeginCreate asynchronous call.
	EndFind	Gets the result from a BeginFind asynchronous call.
	EndGame	Changes the session state from NetworkSessionState.Playing to NetworkSessionState.Lobby .
	EndJoin	Gets the result from a BeginJoin asynchronous call.
	EndJoinInvited	Gets the result from a BeginJoinInvited asynchronous call.
	Equals	(Inherited from Object .)
	Find	Overloaded. Issues a matchmaking query, searching for available multiplayer sessions.
	FindGamerById	Looks up the network gamer with the specified ID.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Join	Joins an existing multiplayer session.
	JoinInvited	Overloaded. Joins the specified local gamers, along with the local machine, to an existing network session in response to an InviteAccepted notification event from a non-local session.
	ReferenceEquals	(Inherited from Object .)
	ResetReady	Resets the IsReady property of all session gamers.
	StartGame	Changes the session state from NetworkSessionState.Lobby to NetworkSessionState.Playing .
	ToString	(Inherited from Object .)
	Update	Updates the state of the multiplayer session.

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.AddLocalGamer Method

Adds the specified local gamer profile to the network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void AddLocalGamer (  
    SignedInGamer gamer  
)
```

Parameters

gamer

The gamer to add to the network session

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginCreate Method

Starts hosting a new multiplayer session.

Overload List

Name	Description
NetworkSession.BeginCreate (NetworkSessionType, Generic IEnumerable, Int32, Int32, NetworkSessionProperties, AsyncCallback, Object)	Starts hosting a new multiplayer session.
NetworkSession.BeginCreate (NetworkSessionType, Int32, Int32, AsyncCallback, Object)	Starts hosting a new multiplayer session.
NetworkSession.BeginCreate (NetworkSessionType, Int32, Int32, Int32, NetworkSessionProperties, AsyncCallback, Object)	Starts hosting a new multiplayer session.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.
InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> NetworkSession.Create NetworkSession.BeginCreate NetworkSession.Find NetworkSession.BeginFind NetworkSession.Join NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.BeginCreate Method (NetworkSessionType, Generic IEnumerable, Int32, Int32, NetworkSessionProperties, AsyncCallback, Object)

Starts hosting a new multiplayer session. The creation operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginCreate (
    NetworkSessionType sessionType,
    IEnumerable<SignedInGamer> localGamers,
    int maxGamers,
    int privateGamerSlots,
    NetworkSessionProperties sessionProperties,
    AsyncCallback callback,
    Object asyncState
)
```

Parameters

sessionType

Type of session being created.

localGamers

A collection of local gamers to add to the network session. The first gamer in the collection becomes the host; the others are added to the session.

maxGamers

Maximum number of players allowed in this network session. For Windows- or Xbox 360-based games, this value must be between 2 and 31; 31 is the maximum number of players supported in a session. For Zune-based games, this value must be between 2 and 8; 8 is the maximum number of players supported in the session.

privateGamerSlots

Number of reserved private session slots created for the session. This value must be less than *maximumGamers*.

sessionProperties

Properties of the session being created.

callback

The method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method.


Exceptions


Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.


InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. • One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> • NetworkSession.Create • NetworkSession.BeginCreate • NetworkSession.Find • NetworkSession.BeginFind • NetworkSession.Join • NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

Remarks

Call [EndCreate](#) to get results of the call to **BeginCreate**.

<p> Zune Specific Information</p>
<p>For Zune-based games, <i>maxGamers</i> must be between 2 and 8; 8 is the maximum number of players supported in the session.</p>

<p> Windows Specific Information</p>
<p>For Windows-based games, <i>maxGamers</i> must be between 2 and 31; 31 is the maximum number of players supported in a session.</p>

<p> Xbox 360 Specific Information</p>
<p>For Xbox 360-based games, <i>maxGamers</i> must be between 2 and 31; 31 is the maximum number of players supported in a session.</p>

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginCreate Method (NetworkSessionType, Int32, Int32, AsyncCallback, Object)

Starts hosting a new multiplayer session. The creation operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginCreate (
    NetworkSessionType sessionType,
    int maxLocalGamers,
    int maxGamers,
    AsyncCallback callback,
    Object asyncState
)
```

Parameters

sessionType

Type of session being created.

maxLocalGamers

Maximum number of local players on the same gaming machine in this network session.

maxGamers

Maximum number of players allowed in this network session. For Windows- or Xbox 360-based games, this value must be between 2 and 31; 31 is the maximum number of players supported in a session. For Zune-based games, this value must be between 2 and 8; 8 is the maximum number of players supported in the session.

callback

The method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.

InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. • One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> • NetworkSession.Create • NetworkSession.BeginCreate • NetworkSession.Find • NetworkSession.BeginFind • NetworkSession.Join • NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

Remarks

Call [EndCreate](#) to get results of the call to **BeginCreate**.

Zune Specific Information

For Zune-based games, *maxGamers* must be between 2 and 8; 8 is the maximum number of players supported in the session.

Windows Specific Information

For Windows-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Xbox 360 Specific Information

For Xbox 360-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginCreate Method (NetworkSessionType, Int32, Int32, Int32, NetworkSessionProperties, AsyncCallback, Object)

Starts hosting a new multiplayer session. The creation operation is performed asynchronously.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginCreate (
    NetworkSessionType sessionType,
    int maxLocalGamers,
    int maxGamers,
    int privateGamerSlots,
    NetworkSessionProperties sessionProperties,
    AsyncCallback callback,
    Object asyncState
)
```

Parameters

sessionType

Type of session being created.

maxLocalGamers

Maximum number of local players on the same gaming machine in this network session.

maxGamers

Maximum number of players allowed in this network session. For Windows- or Xbox 360-based games, this value must be between 2 and 31; 31 is the maximum number of players supported in a session. For Zune-based games, this value must be between 2 and 8; 8 is the maximum number of players supported in the session.

privateGamerSlots

Number of reserved private session slots created for the session. This value must be less than *maximumGamers*.

sessionProperties

Properties of the session being created.

callback

The method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method. Call [EndCreate](#) to access these results.


Exceptions


Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.


InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. • One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> • NetworkSession.Create • NetworkSession.BeginCreate • NetworkSession.Find • NetworkSession.BeginFind • NetworkSession.Join • NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

Remarks

Call [EndCreate](#) to get results of the call to **BeginCreate**.

<p> Zune Specific Information</p> <p>For Zune-based games, <i>maxGamers</i> must be between 2 and 8; 8 is the maximum number of players supported in the session.</p>
--

<p> Windows Specific Information</p> <p>For Windows-based games, <i>maxGamers</i> must be between 2 and 31; 31 is the maximum number of players supported in a session.</p>
--

<p> Xbox 360 Specific Information</p> <p>For Xbox 360-based games, <i>maxGamers</i> must be between 2 and 31; 31 is the maximum number of players supported in a session.</p>
--

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginFind Method

Starts a matchmaking query to search for available multiplayer sessions. Call [EndFind](#) to access the results of this asynchronous call.

Overload List

Name	Description
NetworkSession.BeginFind (NetworkSessionType, IEnumerable, NetworkSessionProperties, AsyncCallback, Object)	Starts a matchmaking query to search for available multiplayer sessions, specifying a collection of local gamers to add to the network session.
NetworkSession.BeginFind (NetworkSessionType, Int32, NetworkSessionProperties, AsyncCallback, Object)	Starts a matchmaking query to search for available multiplayer sessions.

Exceptions

Exception type	Condition
ArgumentException	BeginFind cannot be used from NetworkSession objects of session type NetworkSessionType.Local .
ArgumentOutOfRangeException	<i>maxLocalGamers</i> must be between 1 and 4.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.BeginFind Method (NetworkSessionType, Generic IEnumerable, NetworkSessionProperties, AsyncCallback, Object)

Starts a matchmaking query to search for available multiplayer sessions, specifying a collection of local gamers to add to the network session. Call [EndFind](#) to access the results of this asynchronous call.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginFind (
    NetworkSessionType sessionType,
    IEnumerable<SignedInGamer> localGamers,
    NetworkSessionProperties searchProperties,
    AsyncCallback callback,
    Object asyncState
)
```

Parameters

sessionType

Type of session sought.

localGamers

A collection of local gamers to add to the network session.

searchProperties

Optional session properties to match during the search. If this argument is **null**, the query results will not be filtered based on the session properties of available sessions. If any of the search property values in *searchProperties* are specified, only sessions with matching properties will be returned by the query. **null** items in the *searchProperties* collection will match to any value for that session property.

callback

Method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
ArgumentException	BeginFind cannot be used from NetworkSession objects of session type NetworkSessionType.Local .
ArgumentOutOfRangeException	<i>maxLocalGamers</i> must be between 1 and 4.

RemarksThe collection of gamers provided in *localGamers* are not added to a session until [NetworkSession.Join](#) is called with one of the sessions returned from the search. The gamers are provided here for matchmaking purposes.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginFind Method (NetworkSessionType, Int32, NetworkSessionProperties, AsyncCallback, Object)

Starts a matchmaking query to search for available multiplayer sessions. Call [EndFind](#) to access the results of this asynchronous call.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginFind (
    NetworkSessionType sessionType,
    int maxLocalGamers,
    NetworkSessionProperties searchProperties,
    AsyncCallback callback,
    Object asyncState
)
```

Parameters

sessionType

Type of session sought.

maxLocalGamers

Maximum number of local players on the same gaming machine in this network session.

searchProperties

Optional session properties to match during the search. If this argument is **null**, the query results will not be filtered based on the session properties of available sessions. If any of the search property values in *searchProperties* are specified, only sessions with matching properties will be returned by the query. **null** items in the *searchProperties* collection will match to any value for that session property.

callback

Method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
ArgumentException	BeginFind cannot be used from NetworkSession objects of session type NetworkSessionType.Local .
ArgumentOutOfRangeException	<i>maxLocalGamers</i> must be between 1 and 4.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginJoin Method

Starts a join operation for the specified multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginJoin (  
    AvailableNetworkSession availableSession,  
    AsyncCallback callback,  
    Object asyncState  
)
```

Parameters

availableSession

Session the gamer is attempting to join.

callback

The method to be called once the asynchronous operation has finished.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method. Call [EndJoin](#) to access these results.

Exceptions

Exception type	Condition
ArgumentNullException	<i>availableSession</i> is null .
ObjectDisposedException	<i>availableSession</i> is disposed.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginJoinInvited Method

Starts joining an existing network session in response to an [InviteAccepted](#) notification event. Call [EndJoinInvited](#) to access the asynchronous method results.

Overload List

Name	Description
NetworkSession.BeginJoinInvited (Generic IEnumerable, AsyncCallback, Object)	Starts joining an existing network session in response to an InviteAccepted notification event with the specified collection of local gamers.
NetworkSession.BeginJoinInvited (Int32, AsyncCallback, Object)	Starts joining an existing network session in response to an InviteAccepted notification event.

Remarks

See [How To: Add Support for Game Invitations](#) for more information about adding game invitation support to your title.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.BeginJoinInvited Method (Generic IEnumerable, AsyncCallback, Object)

Starts joining an existing network session in response to an [InviteAccepted](#) notification event with the specified collection of local gamers. Call [EndJoinInvited](#) to access the asynchronous method results.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginJoinInvited (  
    IEnumerable<SignedInGamer> localGamers,  
    AsyncCallback callback,  
    Object asyncState  
)
```

Parameters

localGamers

A collection of local signed-in gamers to add to the network session.

callback

Method to be called at the conclusion of the asynchronous operation.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Remarks

See [How To: Add Support for Game Invitations](#) for more information about adding game invitation support to your title.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BeginJoinInvited Method (Int32, AsyncCallback, Object)

Starts joining an existing network session in response to an [InviteAccepted](#) notification event. Call [EndJoinInvited](#) to access the asynchronous method results.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static IAsyncResult BeginJoinInvited (
    int maxLocalGamers,
    AsyncCallback callback,
    Object asyncState
)
```

Parameters

maxLocalGamers

The number of local signed-in gamers to add to the network session when accepting the invite. This must be at least one, and can be as high as the limit imposed by either the title or platform. On Windows, this value must always be set to one.

callback

Method to be called at the conclusion of the asynchronous operation.

asyncState

State of the asynchronous operation.

Return Value

An [IAsyncResult](#) used to track the progress of the method.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>maxLocalGamers</i> must be between 1 and 4.

Remarks

If there are fewer signed-in gamers than the number specified in *maxLocalGamers*, all currently-available signed-in gamers will be added. This method does not allow specifying particular signed-in gamers in the event that there are more local signed-in gamers than the number to add.

See [How To: Add Support for Game Invitations](#) for more information about adding game invitation support to your title.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Create Method

Hosts a new multiplayer session.

Overload List

Name	Description
NetworkSession.Create (NetworkSessionType, Generic IEnumerable, Int32, Int32, NetworkSessionProperties)	Starts hosting a new multiplayer session.
NetworkSession.Create (NetworkSessionType, Int32, Int32)	Starts hosting a new multiplayer session.
NetworkSession.Create (NetworkSessionType, Int32, Int32, Int32, NetworkSessionProperties)	Starts hosting a new multiplayer session.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.
InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> NetworkSession.Create NetworkSession.BeginCreate NetworkSession.Find NetworkSession.BeginFind NetworkSession.Join NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.Create Method (NetworkSessionType, Generic IEnumerable, Int32, Int32, NetworkSessionProperties)

Starts hosting a new multiplayer session. This method blocks until the operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession Create (
    NetworkSessionType sessionType,
    IEnumerable<SignedInGamer> localGamers,
    int maxGamers,
    int privateGamerSlots,
    NetworkSessionProperties sessionProperties
)
```

Parameters

sessionType

Type of session being hosted.

localGamers

A collection of local gamers to add to the network session. The first gamer in the collection becomes the host; the others are added to the session.

maxGamers

Maximum number of players allowed in this network session. For Windows- or Xbox 360-based games, this value must be between 2 and 31; 31 is the maximum number of players supported in a session. For Zune-based games, this value must be between 2 and 8; 8 is the maximum number of players supported in the session.

privateGamerSlots

Number of reserved private session slots created for the session. This value must be less than *maxGamers*.

sessionProperties

Properties of the session being created.

Return Value

The newly hosted session.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.

InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. • One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> • NetworkSession.Create • NetworkSession.BeginCreate • NetworkSession.Find • NetworkSession.BeginFind • NetworkSession.Join • NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

Remarks

Zune Specific Information

For Zune-based games, *maxGamers* must be between 2 and 8; 8 is the maximum number of players supported in the session.

Windows Specific Information

For Windows-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Xbox 360 Specific Information

For Xbox 360-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Behavior of Local Sessions

XNA Game Studio 3.0 introduces a change in the behavior of network sessions when local profiles sign out in the middle of a session. In previous versions of XNA Game Studio, a session ended whenever a player signed out. Now the session continues even if one or more profiles sign out. Signed-out players are removed from the session, but the session ends only if no suitable players remain in the session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Create Method (NetworkSessionType, Int32, Int32)

Starts hosting a new multiplayer session. This method blocks until the operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession Create (
    NetworkSessionType sessionType,
    int maxLocalGamers,
    int maxGamers
)
```

Parameters

sessionType

Type of session being hosted.

maxLocalGamers

Maximum number of local players on the same gaming machine in this network session.

maxGamers

Maximum number of players allowed in this network session. For Windows- or Xbox 360-based games, this value must be between 2 and 31; 31 is the maximum number of players supported in a session. For Zune-based games, this value must be between 2 and 8; 8 is the maximum number of players supported in the session.

Return Value

The newly hosted session.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.
InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> NetworkSession.Create NetworkSession.BeginCreate NetworkSession.Find NetworkSession.BeginFind NetworkSession.Join NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

Remarks

Zune Specific Information

For Zune-based games, *maxGamers* must be between 2 and 8; 8 is the maximum number of players supported in the session.

Windows Specific Information

For Windows-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Xbox 360 Specific Information

For Xbox 360-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Behavior of Local Sessions

XNA Game Studio 3.0 introduces a change in the behavior of network sessions when local profiles sign out in the middle of a session. In previous versions of XNA Game Studio, a session ended whenever a player signed out. Now the session continues even if one or more profiles sign out. Signed-out players are removed from the session, but the session ends only if no suitable players remain in the session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Create Method (NetworkSessionType, Int32, Int32, Int32, NetworkSessionProperties)

Starts hosting a new multiplayer session. This method blocks until the operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession Create (
    NetworkSessionType sessionType,
    int maxLocalGamers,
    int maxGamers,
    int privateGamerSlots,
    NetworkSessionProperties sessionProperties
)
```

Parameters

sessionType

Type of session being hosted.

maxLocalGamers

Maximum number of local players on the same gaming machine in this network session.

maxGamers

Maximum number of players allowed in this network session. For Windows- or Xbox 360-based games, this value must be between 2 and 31; 31 is the maximum number of players supported in a session. For Zune-based games, this value must be between 2 and 8; 8 is the maximum number of players supported in the session.

privateGamerSlots

Number of reserved private session slots created for the session. This value must be less than *maxGamers*.

sessionProperties

Properties of the session being created.

Return Value

The newly hosted session.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> <i>maxGamers</i> is not in range of gamers that are supported in a network session. The XNA Framework supports multiplayer sessions between 2 and 31 players in a session for a Windows- or an Xbox 360-based game. On Zune, the XNA Framework supports a multiplayer session between 2 and 8 players in a session. <i>privateGamerSlots</i> is less than 0 or greater than <i>maximumGamers</i>.

InvalidOperationException	<p>One of the following conditions is true:</p> <ul style="list-style-type: none"> • A different session already exists. There can only be one <code>NetworkSession</code> or <code>AvailableNetworkSessionCollection</code> in use at a time. You must dispose any previous session before you create a new one. • One of the following methods has been called, but has not yet returned results: <ul style="list-style-type: none"> • NetworkSession.Create • NetworkSession.BeginCreate • NetworkSession.Find • NetworkSession.BeginFind • NetworkSession.Join • NetworkSession.BeginJoin <p>Note that you must wait for any of these methods to return before you can create a new network session. To support this, you may use the asynchronous methods BeginCreate, BeginFind, BeginJoin, and the corresponding EndCreate, EndFind, EndJoin methods to detect when the operation is complete.</p>
ObjectDisposedException	This <code>NetworkSession</code> is disposed.

Remarks

Zune Specific Information

For Zune-based games, *maxGamers* must be between 2 and 8; 8 is the maximum number of players supported in the session.

Windows Specific Information

For Windows-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Xbox 360 Specific Information

For Xbox 360-based games, *maxGamers* must be between 2 and 31; 31 is the maximum number of players supported in a session.

Behavior of Local Sessions

XNA Game Studio 3.0 introduces a change in the behavior of network sessions when local profiles sign out in the middle of a session. In previous versions of XNA Game Studio, a session ended whenever a player signed out. Now the session continues even if one or more profiles sign out. Signed-out players are removed from the session, but the session ends only if no suitable players remain in the session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Dispose Method

Ends the current multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.EndCreate Method

Gets the result from a [BeginCreate](#) asynchronous call.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession EndCreate (  
    IAsyncResult result  
)
```

Parameters

result

An [IAsyncResult](#) used to track the progress of the operation.

Return Value

The session being created.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.EndFind Method

Gets the result from a [BeginFind](#) asynchronous call.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AvailableNetworkSessionCollection EndFind (  
    IAsyncResult result  
)
```

Parameters

result

An [IAsyncResult](#) used to track the progress of the operation.

Return Value

Collection of sessions matching the search criteria.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.EndGame Method

Changes the session state from [NetworkSessionState.Playing](#) to [NetworkSessionState.Lobby](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void EndGame ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	This is disposed.
InvalidOperationException	This NetworkSession is not the host, or is in an invalid state to call this method.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.EndJoin Method

Gets the result from a [BeginJoin](#) asynchronous call.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession EndJoin (  
    IAsyncResult result  
)
```

Parameters

result

An [IAsyncResult](#) used to track the progress of the operation.

Return Value

The session being joined by the gamer.

Exceptions

Exception type	Condition
NetworkSessionJoinException	<p>The session either is full, is not joinable, or cannot be found.</p> <ul style="list-style-type: none">• If the session is full, the specified session does not have enough open slots for all the local gamers signed in to this machine.• If it is not joinable, you must wait for the host to return to the lobby before joining.• If it cannot be found, the session may have ended, or there may be no network connectivity between the local machine and session host.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.EndJoinInvited Method

Gets the result from a [BeginJoinInvited](#) asynchronous call.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession EndJoinInvited (
    IAsyncResult result
)
```

Parameters

result

An [IAsyncResult](#) used to track the progress of the operation.

Return Value

The session being joined by the gamer.

Exceptions

Exception type	Condition
NetworkSessionJoinException	<p>The session either is full, is not joinable, or cannot be found.</p> <ul style="list-style-type: none"> • If the session is full, the specified session does not have enough open slots for all the local gamers signed in to this machine. • If it is not joinable, you must wait for the host to return to the lobby before joining. • If it cannot be found, the session may have ended, or there may be no network connectivity between the local machine and session host.

Remarks

See [How To: Add Support for Game Invitations](#) for more information about adding game invitation support to your title.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method. The object's **Finalize** method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Find Method

Issues a matchmaking query, searching for available multiplayer sessions. This method blocks until the operation has completed.

Overload List

Name	Description
NetworkSession.Find (NetworkSessionType, Generic IEnumerable, NetworkSessionProperties)	Issues a matchmaking query, searching for available multiplayer sessions.
NetworkSession.Find (NetworkSessionType, Int32, NetworkSessionProperties)	Issues a matchmaking query, searching for available multiplayer sessions.

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.Find Method (NetworkSessionType, Generic IEnumerable, NetworkSessionProperties)

Issues a matchmaking query, searching for available multiplayer sessions. This method blocks until the operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AvailableNetworkSessionCollection Find (
    NetworkSessionType sessionType,
    IEnumerable<SignedInGamer> localGamers,
    NetworkSessionProperties searchProperties
)
```

Parameters

sessionType

Type of session sought.

localGamers

A collection of local gamers to add to the network session.

searchProperties

Optional session properties to match during the search. If this argument is **null**, the query results will not be filtered based on the session properties of available sessions. If any of the search property values in *searchProperties* are specified, only sessions with matching properties will be returned by the query. **null** items in the *searchProperties* collection will match to any value for that session property.

Return Value

Collection of available network multiplayer sessions.

Exceptions

Exception type	Condition
ArgumentException	Find cannot be used from NetworkSession objects of session type NetworkSessionType.Local .
ArgumentOutOfRangeException	<i>maxLocalGamers</i> must be between 1 and 4.

RemarksThe collection of gamers provided in *localGamers* are not added to a session until [NetworkSession.Join](#) is called with one of the sessions returned from the search. The gamers are provided here for matchmaking purposes.

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Find Method (NetworkSessionType, Int32, NetworkSessionProperties)

Issues a matchmaking query, searching for available multiplayer sessions. This method blocks until the operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static AvailableNetworkSessionCollection Find (
    NetworkSessionType sessionType,
    int maxLocalGamers,
    NetworkSessionProperties searchProperties
)
```

Parameters

sessionType

Type of session sought.

maxLocalGamers

Maximum number of local players on the same gaming machine in this network session.

searchProperties

Optional session properties to match during the search. If this argument is **null**, the query results will not be filtered based on the session properties of available sessions. If any of the search property values in *searchProperties* are specified, only sessions with matching properties will be returned by the query. **null** items in the *searchProperties* collection will match to any value for that session property.

Return Value

Collection of available network multiplayer sessions.

Exceptions

Exception type	Condition
ArgumentException	Find cannot be used from NetworkSession objects of session type NetworkSessionType.Local .
ArgumentOutOfRangeException	<i>maxLocalGamers</i> must be between 1 and 4.

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.FindGamerById Method

Looks up the network gamer with the specified ID. Returns **null** if the session contains no matching gamer.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkGamer FindGamerById (  
    byte gameId  
)
```

Parameters

gameId

Identifier specifying the network gamer for which to search.

Return Value

Network gamer matching the requested ID, or **null** if no matching gamer was found.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Join Method

Joins an existing multiplayer session. This method blocks until the operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession Join (
    AvailableNetworkSession availableSession
)
```

Parameters

availableSession

An existing multiplayer session.

Return Value

Session joined by the gamer.

Exceptions

Exception type	Condition
ArgumentNullException	<i>availableSession</i> is null .
ObjectDisposedException	<i>availableSession</i> is disposed.
NetworkSessionJoinException	The session either is full, is not joinable, or cannot be found. <ul style="list-style-type: none"> • If the session is full, the specified session does not have enough open slots for all the local gamers signed in to this machine. • If it is not joinable, you must wait for the host to return to the lobby before joining. • If it cannot be found, the session may have ended, or there may be no network connectivity between the local machine and session host.

Remarks

Use the [Find](#) or [BeginFind](#) methods to search for available sessions.

Behavior of Local Sessions

XNA Game Studio 3.0 introduces a change in the behavior of network sessions when local profiles sign out in the middle of a session. In previous versions of XNA Game Studio, a session ended whenever a player signed out. Now the session continues even if one or more profiles sign out. Signed-out players are removed from the session, but the session ends only if no suitable players remain in the session.

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.JoinInvited Method

Joins the specified local gamers, along with the local machine, to an existing network session in response to an [InviteAccepted](#) notification event from a non-local session.

Note

Use the [NetworkSession.AddLocalGamer](#) method to join players to a session in response to an [InviteAccepted](#) notification event for a session that already exists locally, as indicated by [IsCurrentSession](#).

Overload List

Name	Description
NetworkSession.JoinInvited (Generic IEnumerable)	Joins the specified local gamers, along with the local machine, to an existing network session in response to an InviteAccepted notification event from a non-local session.
NetworkSession.JoinInvited (Int32)	Joins the local machine to an existing network session in response to an InviteAccepted notification event from a non-local session.

Exceptions

Exception type	Condition
NetworkSessionJoinException	<p>The session either is full, is not joinable, or cannot be found.</p> <ul style="list-style-type: none"> • If the session is full, the specified session does not have enough open slots for all the local gamers signed in to this machine. • If it is not joinable, you must wait for the host to return to the lobby before you can join it. • If it cannot be found, the session may have ended, or there may be no network connectivity between the local machine and session host.

Remarks

This method blocks until the operation has completed. For an asynchronous version of this method, see [BeginJoinInvited](#).

See Also

Reference

[Guide.ShowGameInvite Method](#)
[NetworkSession.InviteAccepted Event](#)
[NetworkSession.BeginJoinInvited Method](#)
[NetworkSession.EndJoinInvited Method](#)
[NetworkSession Class](#)
[NetworkSession Members](#)
[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.JoinInvited Method (Generic IEnumerable)

Joins the specified local gamers, along with the local machine, to an existing network session in response to an [InviteAccepted](#) notification event from a non-local session.

Note

Use the [NetworkSession.AddLocalGamer](#) method to join players to a session in response to an [InviteAccepted](#) notification event for a session that already exists locally, as indicated by [IsCurrentSession](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession JoinInvited (
    IEnumerable<SignedInGamer> localGamers
)
```

Parameters

localGamers

A collection of local signed-in gamers to add to the network session.

Return Value

The network session.

Exceptions

Exception type	Condition
NetworkSessionJoinException	<p>The session either is full, is not joinable, or cannot be found.</p> <ul style="list-style-type: none"> • If the session is full, the specified session does not have enough open slots for all the local gamers signed in to this machine. • If it is not joinable, you must wait for the host to return to the lobby before you can join it. • If it cannot be found, the session may have ended, or there may be no network connectivity between the local machine and session host.

Remarks

This method blocks until the operation has completed. For an asynchronous version of this method, see [BeginJoinInvited](#).

See Also

Reference

[Guide.ShowGameInvite Method](#)

[NetworkSession.InviteAccepted Event](#)

[NetworkSession.BeginJoinInvited Method](#)

[NetworkSession.EndJoinInvited Method](#)

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.JoinInvited Method (Int32)

Joins the local machine to an existing network session in response to an [InviteAccepted](#) notification event from a non-local session.

Note

Use the [NetworkSession.AddLocalGamer](#) method to join players to a session in response to an [InviteAccepted](#) notification event for a session that already exists locally, as indicated by [IsCurrentSession](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static NetworkSession JoinInvited (
    int maxLocalGamers
)
```

Parameters

maxLocalGamers

The number of local signed-in gamers to add to the network session when accepting the invite. This must be at least one, and can be as high as the limit imposed by either the title or platform. On Windows, this value must always be set to one.

Return Value

The network session.

Exceptions

Exception type	Condition
NetworkSessionJoinException	<p>The session either is full, is not joinable, or cannot be found.</p> <ul style="list-style-type: none"> If the session is full, the specified session does not have enough open slots for all the local gamers signed in to this machine. If it is not joinable, you must wait for the host to return to the lobby before joining. If it cannot be found, the session may have ended, or there may be no network connectivity between the local machine and session host.

Remarks

If there are fewer signed-in gamers than the number specified in *maxLocalGamers*, all currently-available signed-in gamers will be added. This method does not allow specifying particular signed-in gamers in the event that there are more local signed-in gamers than the number to add.

This method blocks until the operation has completed. For an asynchronous version of this method, see [BeginJoinInvited](#).

See Also

Reference

[Guide.ShowGameInvite Method](#)

[NetworkSession.InviteAccepted Event](#)

[NetworkSession.BeginJoinInvited Method](#)

[NetworkSession.EndJoinInvited Method](#)

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.ResetReady Method

Resets the [IsReady](#) property of all session gamers.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void ResetReady ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	This is disposed.
InvalidOperationException	This NetworkSession is not the host, or is in an invalid state to call this method.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.StartGame Method

Changes the session state from [NetworkSessionState.Lobby](#) to [NetworkSessionState.Playing](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void StartGame ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	This is disposed.
InvalidOperationException	This NetworkSession is not the host, or is in an invalid state to call this method.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Update Method

Updates the state of the multiplayer session. Call this method at regular intervals—for example, from the [Game.Update](#) method.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Update ()
```

Exceptions

Exception type	Condition
ObjectDisposedException	This object has already been disposed.

See Also

Reference

[NetworkSession Class](#)




















[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession Properties

Public Properties

	Name	Description
	AllGamers	Gets the collection of gamers currently in the session.
	AllowHostMigration	Gets or sets whether host migration is allowed. This can be read by any machine in the session, but can only be changed by the host. The default value is false indicating that host migration is disabled.
	AllowJoinInProgress	Gets or sets whether join-in-progress is allowed. If the host enables this setting, new machines will be able to join at any time. The default value is false, indicating that join-in-progress is disabled. AllowJoinInProgress can be read by any machine in the session, but can only be changed by the host.
	BytesPerSecondReceived	Gets a performance counter recording the amount of data being received from the network.
	BytesPerSecondSent	Gets a performance counter recording the amount of data being sent over the network.
	Host	Gets the current host of the multiplayer session.
	IsDisposed	Gets a value that indicates whether the object is disposed.
	IsEveryoneReady	Determines whether all gamers are ready to enter the session.
	IsHost	Determines whether this machine is the session host.
	LocalGamers	Get the collection of local gamers for a multiplayer session.
	MaxGamers	Gets or sets the maximum number of players able to join this multiplayer session.
	PreviousGamers	A collection of previous gamers in the network session.
	PrivateGamerSlots	Gets or sets the number of private slots reserved for gamers who join using an invitation.
	RemoteGamers	Gets the collection of remote gamers for a multiplayer session.
	SessionProperties	Gets any custom properties that have been attached to the session.
	SessionState	Gets the current state of a multiplayer session.
	SessionType	Gets the current multiplayer session type.
	SimulatedLatency	Gets or sets the amount of simulated network latency.
	SimulatedPacketLoss	Gets or sets the amount of simulated packet loss.

See Also

Reference

[NetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.AllGamers Property

Gets the collection of gamers currently in the session. This is the union of everything in [LocalGamers](#) and [RemoteGamers](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerCollection<NetworkGamer> AllGamers { get; }
```

Property Value

Collection of gamers in the session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.AllowHostMigration Property

Gets or sets whether host migration is allowed. This can be read by any machine in the session, but can only be changed by the host. The default value is **false** indicating that host migration is disabled.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AllowHostMigration { get; set; }
```

Property Value

true to enable host migration; **false** to disable host migration. The default value is **false**.

Exceptions

Exception type	Condition
InvalidOperationException	This session is not the host, and cannot change the value of AllowHostMigration .
ObjectDisposedException	The session has been disposed.

RemarksWhen host migration is disabled (the default state), the session will be ended if the host leaves the game or there is a network disconnect. If the host enables this setting, after they leave or disconnect a new host will automatically be selected from the machines remaining in the session, and the HostChanged event will be raised to notify you of the change. Note that game simulation state is not automatically transferred during the migration process. It is usually relatively easy for peer-to-peer games to support host migration, but rather more difficult to make this work in a client/server architecture.

See Also

Concepts

[Network Topologies and Host Migration](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.AllowJoinInProgress Property

Gets or sets whether join-in-progress is allowed. If the host enables this setting, new machines will be able to join at any time. The default value is **false**, indicating that join-in-progress is disabled. **AllowJoinInProgress** can be read by any machine in the session, but can only be changed by the host.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool AllowJoinInProgress { get; set; }
```

Property Value

true to enable join-in-progress, **false** otherwise. The default value is **false**.

Exceptions

Exception type	Condition
NotSupportedException	Join-in-progress is not supported for multiplayer sessions of type NetworkSessionType.Ranked .
InvalidOperationException	This session is not the host, and cannot change the value of AllowJoinInProgress .
ObjectDisposedException	The session has been disposed.

Remarks When join-in-progress is disabled (the default state), new machines are only able to join while the session is in the lobby. The session will not be visible to matchmaking queries while in the actual gameplay state.

See Also

Tasks

[How To: Join an In-Progress Game](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BytesPerSecondReceived Property

Gets a performance counter recording the amount of data being received from the network.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BytesPerSecondReceived { get; }
```

Property Value

A performance counter recording the amount of data being received from the network.

Remarks Developers can use this to monitor how much network bandwidth their game is using.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.BytesPerSecondSent Property

Gets a performance counter recording the amount of data being sent over the network.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BytesPerSecondSent { get; }
```

Property Value

A performance counter recording the amount of data being sent over the network.

RemarksDevelopers can use this to monitor how much network bandwidth their game is using.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.Host Property

Gets the current host of the multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkGamer Host { get; }
```

Property Value

The gamer currently hosting the session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.IsEveryoneReady Property

Determines whether all gamers are ready to enter the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsEveryoneReady { get; }
```

Property Value

true if all gamers are ready to leave the lobby screen and begin gameplay. This state is indicated by a setting of **true** for the [IsReady](#) property.

See Also

Tasks

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Reference

[IsReady](#)

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.IsHost Property

Determines whether this machine is the session host.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsHost { get; }
```

Property Value

true if this machine is the session host; **false** otherwise.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.LocalGamers Property

Get the collection of local gamers for a multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerCollection<LocalNetworkGamer> LocalGamers { get; }
```

Property Value

Collection of local gamers of a network session. For a complete collection of gamers, access the [AllGamers](#) property.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.MaxGamers Property

Gets or sets the maximum number of players able to join this multiplayer session. The Zune platform supports up to eight players. All other platforms support up to 31 players.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int MaxGamers { get; set; }
```

Property Value

Maximum number of players supported by the multiplayer session. Only the host can modify this value.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<code>MaxGamers</code> must be between 2 and 31 players for Windows-based and Xbox 360–based games, and between 2 and 8 for Zune-based games.
ObjectDisposedException	This NetworkSession is disposed.
InvalidOperationException	This NetworkSession is not the host; therefore, it cannot set the property MaxGamers .

Remarks

Zune Specific Information

For Zune-based games, **MaxGamers** must be between 2 and 8, where 8 is the maximum number of players supported in the session.

Windows Specific Information

For Windows-based games, **MaxGamers** must be between 2 and 31, where 31 is the maximum number of players supported in a session.

Xbox 360 Specific Information

For Xbox 360–based games, **MaxGamers** must be between 2 and 31, where 31 is the maximum number of players supported in a session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.PreviousGamers Property

A collection of previous gamers in the network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerCollection<NetworkGamer> PreviousGamers { get; }
```

Property Value

A collection of gamers who once belonged to the current network session, but have since left.

RemarksThe number of gamers in the **PreviousGamers** collection will be equal to or less than the value of [MaxPreviousGamers](#). You can use this collection to include previous gamers in Guide calls that take a [Gamer](#) parameter.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.PrivateGamerSlots Property

Gets or sets the number of private slots reserved for gamers who join using an invitation.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int PrivateGamerSlots { get; set; }
```

Property Value

Number of private slots.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	There are not enough slots available to support this new value.
ObjectDisposedException	This NetworkSession is disposed.
InvalidOperationException	This NetworkSession is not the host, therefore it cannot set the property PrivateGamerSlots .

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.RemoteGamers Property

Gets the collection of remote gamers for a multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public GamerCollection<NetworkGamer> RemoteGamers { get; }
```

Property Value

Collection of remote gamers of a network session. For a complete collection of gamers, access the [AllGamers](#) property.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.SessionProperties Property

Gets any custom properties that have been attached to the session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionProperties SessionProperties { get; }
```

Property Value

Any custom properties that have been attached to the session.

RemarksThis [NetworkSessionProperties](#) is automatically synchronized across all the machines in the session, and can only be modified by the host. It can be used to filter sessions during matchmaking operations by passing in a set of required property values to [Find](#) or [BeginFind](#).

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.SessionState Property

Gets the current state of a multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionState SessionState { get; }
```

Property Value

Current state of the session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.SessionType Property

Gets the current multiplayer session type.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionType SessionType { get; }
```

Property Value

The type of the multiplayer session.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.SimulatedLatency Property

Gets or sets the amount of simulated network latency.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan SimulatedLatency { get; set; }
```

Property Value

The amount of simulated network latency, in milliseconds. The default value is 0.

Remarks

SimulatedLatency is useful if you are testing a system link game on a fast local network and want to make sure it will also work correctly over the Internet. Xbox games are typically expected to work with up to 200 milliseconds of latency.

The latency simulation is applied on the sending machine, so if you set this property differently on each machine, only outgoing packets will be affected by the local value of the property. Latency introduced through this setting is not included in the [RoundtripTime](#) property, which always just reports the physical network latency.

Packets sent without any ordering guarantee will be given a random latency normally distributed around the specified value, introducing packet reordering as well as raw latency. Packets sent using [SendDataOptions.InOrder](#) will be delayed without reordering.

⚠ Caution

The packet loss and latency simulation in the XNA Framework applies to messages that are sent to all players, including local players. In game designs where data to local players is sent and received using the network session [SendData](#) methods, local players will experience the set values for simulated packet loss and latency. To work around this issue, avoid broadcasting messages to all users and set [SimulatedPacketLoss](#) and [SimulatedLatency](#) on a per-send basis.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.SimulatedPacketLoss Property

Gets or sets the amount of simulated packet loss.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public float SimulatedPacketLoss { get; set; }
```

Property Value

The amount of simulated packet loss, expressed as any fractional probability in the range 0 to 1, with 1 causing 100% packet loss. The default value is zero.

Remarks

SimulatedPacketLoss is useful if you are testing a system link game on a fast local network and want to make sure it will also work correctly over the Internet. Xbox games are typically expected to work with up to 0.1 (10%) packet loss.

The packet loss simulation is applied on the sending machine, so if you set this property differently on each machine, only outgoing packets will be affected by the local value of the property.

Packets sent using [SendDataOptions.Reliable](#) are not affected by this setting.

⚠ Caution

The packet loss and latency simulation in the XNA Framework applies to messages that are sent to all players, including local players. In game designs where data to local players is sent and received using the network session [SendData](#) methods, local players will experience the set values for simulated packet loss and latency. To work around this issue, avoid broadcasting messages to all users and set [SimulatedPacketLoss](#) and [SimulatedLatency](#) on a per-send basis.

See Also

Reference

[NetworkSession Class](#)









[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession Events

Public Events

	Name	Description
	GameEnded	Occurs when the game moves from gameplay to the lobby.
	GamerJoined	Occurs when a new player joins a multiplayer session.
	GamerLeft	Occurs when a player leaves the multiplayer session.
	GameStarted	Occurs when the game moves from the lobby into actual gameplay.
	HostChanged	Occurs when the session host has changed.
 	InviteAccepted	Event that occurs when a user has accepted an invitation to join a network session.
	SessionEnded	Occurs when the multiplayer session ends.

See Also

Reference

[NetworkSession Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSession.GameEnded Event

Occurs when the game moves from gameplay to the lobby.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<GameEndedEventArgs> GameEnded
```

See Also

Tasks

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.GamerJoined Event

Occurs when a new player joins a multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<GamerJoinedEventArgs> GamerJoined
```

Remarks

This event is raised for both local and remote players, including when the session is first created.

See Also

Tasks

[How To: Manage Players Joining and Leaving the Game](#)

[How To: Join an In-Progress Game](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.GamerLeft Event

Occurs when a player leaves the multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<GamerLeftEventArgs> GamerLeft
```

RemarksThis event is raised for both local and remote gamers.

See Also

Tasks

[How To: Manage Players Joining and Leaving the Game](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.GameStarted Event

Occurs when the game moves from the lobby into actual gameplay.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<GameStartedEventArgs> GameStarted
```

See Also

Tasks

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.HostChanged Event

Occurs when the session host has changed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<HostChangedEventArgs> HostChanged
```

Remarks **HostChanged** is only raised during host migration, and not when the session is first created.

See Also

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.InviteAccepted Event

Event that occurs when a user has accepted an invitation to join a network session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler<InviteAcceptedEventArgs> InviteAccepted
```

Remarks

In addition to being sent if a player responds to a game invitation, this event is also sent to a title when a gamer requests to join an in-progress session by selecting **Join Session In Progress** from the gamer's friends list on the LIVE Guide.

Important

If your title does not register for the [InviteAccepted](#) event, its sessions will always show up as non-joinable.

Notification of this event may be delivered at any time. However, if the game has been launched in response to a cross-title invite, it will arrive as soon as the game starts. In this case, the event will be delivered as soon as the managed client subscribes to it.

In response to this event, games should end any existing network session and call [JoinInvited](#). There is no need to display any user interface before this. The system will already have prompted the user to confirm acceptance of the game invitation.

For more information, see [How To: Add Support for Game Invitations](#).

See Also

Reference

[Guide.ShowGameInvite Method](#)

[NetworkSession.JoinInvited Method](#)

[NetworkSession.BeginJoinInvited Method](#)

[NetworkSession.EndJoinInvited Method](#)

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSession.SessionEnded Event

Occurs when the multiplayer session ends.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler<NetworkSessionEndedEventArgs> SessionEnded
```

See Also

Tasks

[How To: Manage Player Movement Between Lobby and Gameplay Modes](#)

Reference

[NetworkSession Class](#)

[NetworkSession Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionEndedEventArgs Class

Represents the arguments passed to a [SessionEnded](#) event. These arguments are passed to event handlers when a session ends.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class NetworkSessionEndedEventArgs : EventArgs
```

See Also

Reference

[NetworkSessionEndedEventArgs Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune


NetworkSessionEndedEventArgs Members

The following tables list the members exposed by the NetworkSessionEndedEventArgs type.






Public Constructors

Name	Description
 NetworkSessionEndedEventArgs	Creates an instance of NetworkSessionEndedEventArgs .



Public Properties

Name	Description
 EndReason	Gets the reason for ending a session.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkSessionEndedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionEndedEventArgs Constructor

Creates an instance of [NetworkSessionEndedEventArgs](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionEndedEventArgs (  
    NetworkSessionEndReason endReason  
)
```

Parameters

endReason

Reason for ending the session.

See Also

Reference

[NetworkSessionEndedEventArgs Class](#)






[NetworkSessionEndedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionEndedEventArgs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[NetworkSessionEndedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionEndedEventArgs Properties

Public Properties

	Name	Description
	EndReason	Gets the reason for ending a session.

See Also

Reference

[NetworkSessionEndedEventArgs Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionEndedEventArgs.EndReason Property

Gets the reason for ending a session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionEndReason EndReason { get; }
```

Property Value

Reason for ending the session.

See Also

Reference

[NetworkSessionEndedEventArgs Class](#)

[NetworkSessionEndedEventArgs Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionEndReason Enumeration

Defines the reason a session ended.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum NetworkSessionEndReason
```

Members

Member name	Description
ClientSignedOut	This client player has signed out of session.
HostEndedSession	The host left the session, removing all active players.
RemovedByHost	The host removed this client player from the session.
Disconnected	Network connectivity problems ended the session

See Also

Reference

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinError Enumeration

Contains additional data about a [NetworkSessionJoinException](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum NetworkSessionJoinError
```

Members

Member name	Description
SessionNotFound	The session could not be found. Occurs if the session has ended after the matchmaking query but before the client joined, or if there is no network connectivity between the client and session host machines.
SessionNotJoinable	The session exists but is not joinable. Occurs if the session is in progress but does not allow gamers to join a session in progress.
SessionFull	The session exists but does not have any open slots for local signed-in gamers.

See Also

Reference

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinException Class

Thrown if an error was encountered while joining a session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[SerializableAttribute]  
public class NetworkSessionJoinException : NetworkException
```

See Also

Reference

[NetworkSessionJoinException Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









NetworkSessionJoinException Members

The following tables list the members exposed by the NetworkSessionJoinException type.


Public Constructors

Name	Description
 NetworkSessionJoinException	Overloaded. Creates an instance of NetworkSessionJoinException.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 JoinError	Gets or sets a more detailed description of the session join failure.
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	When overridden in a derived class, returns information about the exception.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkSessionJoinException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionJoinException Constructor

Creates an instance of **NetworkSessionJoinException**.

Overload List

Name	Description
NetworkSessionJoinException ()	Creates an empty instance of NetworkSessionJoinException .
NetworkSessionJoinException (SerializationInfo, StreamingContext)	Creates an instance of NetworkSessionJoinException with the specified streaming context.
NetworkSessionJoinException (String)	Creates an instance of NetworkSessionJoinException with the specified error message.
NetworkSessionJoinException (String, Exception)	Creates an instance of NetworkSessionJoinException with the specified error message and inner exception.
NetworkSessionJoinException (String, NetworkSessionJoinError)	Creates an instance of NetworkSessionJoinException with the specified error and additional information on the exception.

See Also

Reference

[NetworkSessionJoinException Class](#)

[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionJoinException Constructor ()

Creates an empty instance of **NetworkSessionJoinException**.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionJoinException ()
```

See Also

Reference

[NetworkSessionJoinException Class](#)

[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Creates an instance of **NetworkSessionJoinException** with the specified streaming context.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected NetworkSessionJoinException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Describes information related to the session being joined when the exception occurred.

context

Describes the stream where the exception occurred.

See Also

Reference

[NetworkSessionJoinException Class](#)

[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NetworkSessionJoinException Constructor (String)

Creates an instance of **NetworkSessionJoinException** with the specified error message.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionJoinException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[NetworkSessionJoinException Class](#)

[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinException Constructor (String, Exception)

Creates an instance of **NetworkSessionJoinException** with the specified error message and inner exception.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionJoinException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The inner exception related to this exception.

See Also

Reference

[NetworkSessionJoinException Class](#)

[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinException Constructor (String, NetworkSessionJoinError)

Creates an instance of **NetworkSessionJoinException** with the specified error and additional information on the exception.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionJoinException (  
    string message,  
    NetworkSessionJoinError joinError  
)
```

Parameters

message

A message that describes the error.

joinError

Reason for the exception.

See Also

Reference

[NetworkSessionJoinException Class](#)







[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	When overridden in a derived class, returns information about the exception.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NetworkSessionJoinException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionJoinException.GetObjectData Method

When overridden in a derived class, returns information about the exception. In addition to the base behavior, this method provides serialization functionality.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override void GetObjectData (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Information necessary for serialization and deserialization of the session item.

context

Information necessary for the source and destination of a given serialized stream. Also provides an additional caller-defined context.

See Also

Reference

[NetworkSessionJoinException Class](#)









[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)


Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionJoinException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	JoinError	Gets or sets a more detailed description of the session join failure.
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[NetworkSessionJoinException Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionJoinException.JoinError Property

Gets or sets a more detailed description of the session join failure.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionJoinError JoinError { get; set; }
```

Property Value

Contains information about the session join failure.

See Also

Reference

[NetworkSessionJoinException Class](#)

[NetworkSessionJoinException Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionProperties Class

Describes custom, game-specific information about a [NetworkSession](#) object.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class NetworkSessionProperties
```

Remarks

Examples of custom properties include specifying the current game mode or the current level selection. Use these properties to filter the results from a search using the [Find](#) or [BeginFind](#) methods.

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[NetworkSessionProperties Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune



NetworkSessionProperties Members

The following tables list the members exposed by the NetworkSessionProperties type.







Public Constructors

Name	Description
 NetworkSessionProperties	Creates an empty instance of NetworkSessionProperties.



Public Properties

Name	Description
 Count	Gets the number of custom session properties.
 Item	Gets or sets a custom session property value at the specified index











Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator for iterating over the custom property values.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.IsReadOnly	Gets a value indicating whether the collection is read-only.
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add	The System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear	The System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Contains	Checks whether the collection contains the specified value.
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.CopyTo	Copies the contents of the collection to an array.
 System.Collections.IEnumerable.GetEnumerator	Gets an enumerator for iterating over the custom property values.
 System.Collections.Generic.IList<System.Nullable<System.Int32>>.IndexOf	Gets the index of the specified value.
 System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert	The System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt	The System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove	The System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove interface method is not supported by NetworkSessionProperties .

See Also

Tasks

[How To: Find and Join a Network Session](#)

Reference

[NetworkSessionProperties Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionProperties Constructor

Creates an empty instance of **NetworkSessionProperties**.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public NetworkSessionProperties ()
```

See Also

Reference

[NetworkSessionProperties Class](#)







[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionProperties Methods











Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator for iterating over the custom property values.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.IsReadOnly	Gets a value indicating whether the collection is read-only.
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add	The System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear	The System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Contains	Checks whether the collection contains the specified value.
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.CopyTo	Copies the contents of the collection to an array.
 System.Collections.IEnumerable.GetEnumerator	Gets an enumerator for iterating over the custom property values.
 System.Collections.Generic.IList<System.Nullable<System.Int32>>.IndexOf	Gets the index of the specified value.
 System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert	The System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt	The System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt interface method is not supported by NetworkSessionProperties .
 System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove	The System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove interface method is not supported by NetworkSessionProperties .

See Also

Reference

[NetworkSessionProperties Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionProperties.GetEnumerator Method

Gets an enumerator for iterating over the custom property values.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public IEnumerator<Nullable<int>> GetEnumerator ()
```

Return Value

An enumerator for iterating over the custom property values.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add Method

The **System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add** interface method is not supported by [NetworkSessionProperties](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Add (
    Nullable<int> item
)
```

Parameters

item

The value to add.

Exceptions

Exception type	Condition
NotSupportedException	This interface method is not supported by the NetworkSessionProperties type.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear Method

The **System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear** interface method is not supported by [NetworkSessionProperties](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Clear ()
```

Exceptions

Exception type	Condition
NotSupportedException	This interface method is not supported by the NetworkSessionProperties type.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Contains Method

Checks whether the collection contains the specified value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Contains (
    Nullable<int> item
)
```

Parameters

item

The value to search for.

Return Value

true if the collection contains the specified value; **false** otherwise.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.CopyTo Method

Copies the contents of the collection to an array.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.CopyTo (
    Nullable<int>[] array,
    int arrayIndex
)
```

Parameters

array

The array to receive the contents of the collection.

arrayIndex

Starting index for the copy operation.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove Method

The **System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove** interface method is not supported by [NetworkSessionProperties](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.Remove (
    Nullable<int> item
)
```

Parameters

item

The item to remove.

Return Value

true if the item was removed successfully; **false** otherwise.

Exceptions

Exception type	Condition
NotSupportedException	This interface method is not supported by the NetworkSessionProperties type.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.IList<System.Nullable<System.Int32>>.IndexOf Method

Gets the index of the specified value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private int System.Collections.Generic.IList<System.Nullable<System.Int32>>.IndexOf (
    Nullable<int> item
)
```

Parameters

item

The value to search for.

Return Value

The index of the specified value.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert Method

The **System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert** interface method is not supported by [NetworkSessionProperties](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<System.Nullable<System.Int32>>.Insert (  
    int index,  
    Nullable<int> item  
)
```

Parameters

index

Index for the insertion operation

item

Item to insert.

Exceptions

Exception type	Condition
NotSupportedException	This interface method is not supported by the NetworkSessionProperties type.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt Method

The **System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt** interface method is not supported by [NetworkSessionProperties](#).

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<System.Nullable<System.Int32>>.RemoveAt (  
    int index  
)
```

Parameters

index

Index of the item to remove.

Exceptions

Exception type	Condition
NotSupportedException	This interface method is not supported by the NetworkSessionProperties type.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

System.Collections.IEnumerable.GetEnumerator Method

Gets an enumerator for iterating over the custom property values.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

An enumerator for iterating over the custom property values.

See Also

Reference

[NetworkSessionProperties Class](#)



[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionProperties Properties

Public Properties

	Name	Description
	Count	Gets the number of custom session properties.
	Item	Gets or sets a custom session property value at the specified index

See Also

Reference

[NetworkSessionProperties Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

NetworkSessionProperties.Count Property

Gets the number of custom session properties.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

The number of custom session properties.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionProperties.Item Property

Gets or sets a custom session property value at the specified index

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Nullable<int> this [
    int index
] { get; set; }
```

Property Value

The custom session property value.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>index</i> was out of range.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionProperties.System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.IsReadOnly Property

Gets a value indicating whether the collection is read-only.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<System.Nullable<System.Int32>>.IsReadOnly { get; }
```

Property Value

true if the collection is read-only; **false** otherwise.

See Also

Reference

[NetworkSessionProperties Class](#)

[NetworkSessionProperties Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PlatformsXbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionState Enumeration

Defines the different states of a multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum NetworkSessionState
```

Members

Member name	Description
Lobby	The local machine joins the session, waiting in the pregame lobby. The GameStarted event is raised when gameplay begins.
Playing	The local machine joins the session, currently in the middle of gameplay. The GameEnded event is raised when the session returns to the lobby.
Ended	The local machine has left the current session or the session has ended. The SessionEnded event is raised at this time. The event's arguments describe the reason for the session ending.

See Also

Reference

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

NetworkSessionType Enumeration

Defines the different types of a multiplayer session.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public enum NetworkSessionType
```

Members

Member name	Description
Local	Does not involve any networking traffic, but can be used for split-screen gaming on a single Xbox 360 console. Creating a local network session may also make it easier to share code between local and online game modes.
SystemLink	Connect multiple Xbox 360 consoles or computers over a local subnet. These machines do not require a connection to Xbox LIVE or any LIVE accounts. However, connection to machines on different subnets is not allowed. If you are a Creators Club developer testing your game, you can use this type to connect an Xbox 360 console to a computer. However, cross-platform networking is not supported in games distributed to non-Creators Club community players.
PlayMatch	Uses the Xbox LIVE servers. This enables connection to other machines over the Internet. It requires a LIVE Silver Membership for Windows-based games or a LIVE Gold membership for Xbox 360 games. Games in development will also require an XNA Creators Club premium membership. While in trial mode, Indie games downloaded from Xbox LIVE Marketplace will not have access to LIVE matchmaking.
Ranked	All session matches are ranked. This option is available only for commercial games that have passed Xbox LIVE certification. Due to the competitive nature of the gameplay, this session type does not support join-in-progress.

See Also

Tasks

[How To: Join an In-Progress Game](#)

Reference

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader Class

Provides common functionality for efficiently reading incoming network packets.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class PacketReader : BinaryReader
```

Remarks

Commonly, your multiplayer game should create a single **PacketReader** instance at startup and then reuse it whenever a packet needs to be read. To read the packet, pass the **PacketReader** instance to [ReceiveData](#) and then use various Read methods to extract the data.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




PacketReader Members

The following tables list the members exposed by the PacketReader type.































Public Constructors

Name	Description
 PacketReader	Overloaded. Initializes a new instance of PacketReader.





Public Properties

Name	Description
 BaseStream	(Inherited from BinaryReader .)
 Length	Gets the length of the packet being read.
 Position	Gets or sets the current packet read position.

Public Methods

Name	Description
 Close	(Inherited from BinaryReader .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 PeekChar	(Inherited from BinaryReader .)
 Read	(Inherited from BinaryReader .)
 ReadBoolean	(Inherited from BinaryReader .)
 ReadByte	(Inherited from BinaryReader .)
 ReadBytes	(Inherited from BinaryReader .)
 ReadChar	(Inherited from BinaryReader .)
 ReadChars	(Inherited from BinaryReader .)
 ReadColor	Reads a Color value.
 ReadDecimal	(Inherited from BinaryReader .)
 ReadDouble	Reads an 8-byte floating point value.
 ReadInt16	(Inherited from BinaryReader .)
 ReadInt32	(Inherited from BinaryReader .)
 ReadInt64	(Inherited from BinaryReader .)
 ReadMatrix	Reads a Matrix value.
 ReadQuaternion	Reads a Quaternion value.
 ReadSByte	(Inherited from BinaryReader .)
 ReadSingle	Reads a 4-byte floating point value.
 ReadString	(Inherited from BinaryReader .)
 ReadUInt16	(Inherited from BinaryReader .)
 ReadUInt32	(Inherited from BinaryReader .)
 ReadUInt64	(Inherited from BinaryReader .)
 ReadVector2	Reads a Vector2 value.
 ReadVector3	Reads a Vector3 value.
 ReadVector4	Reads a Vector3 value.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Dispose	(Inherited from BinaryReader .)
 FillBuffer	(Inherited from BinaryReader .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Tasks

[How To: Receive Data](#)

Reference

PacketReader Class
Microsoft.Xna.Framework.Net Namespace

PacketReader Constructor

Initializes a new instance of **PacketReader**.

Overload List

Name	Description
PacketReader ()	Initializes an empty instance of PacketReader .
PacketReader (Int32)	Initializes an empty instance of PacketReader with the specified options.

See Also

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketReader Constructor ()

Initializes an empty instance of **PacketReader**.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PacketReader ()
```

See Also

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader Constructor (Int32)

Initializes an empty instance of **PacketReader** with the specified options.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PacketReader (  
    int capacity  
)
```

Parameters

capacity

Initial capacity for a received network packet.

See Also

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader Methods

Public Methods

Name	Description
Close	(Inherited from BinaryReader .)
Equals	(Inherited from Object .)
GetHashCode	(Inherited from Object .)
GetType	(Inherited from Object .)
PeekChar	(Inherited from BinaryReader .)
Read	(Inherited from BinaryReader .)
ReadBoolean	(Inherited from BinaryReader .)
ReadByte	(Inherited from BinaryReader .)
ReadBytes	(Inherited from BinaryReader .)
ReadChar	(Inherited from BinaryReader .)
ReadChars	(Inherited from BinaryReader .)
ReadColor	Reads a Color value.
ReadDecimal	(Inherited from BinaryReader .)
ReadDouble	Reads an 8-byte floating point value.
ReadInt16	(Inherited from BinaryReader .)
ReadInt32	(Inherited from BinaryReader .)
ReadInt64	(Inherited from BinaryReader .)
ReadMatrix	Reads a Matrix value.
ReadQuaternion	Reads a Quaternion value.
ReadSByte	(Inherited from BinaryReader .)
ReadSingle	Reads a 4-byte floating point value.
ReadString	(Inherited from BinaryReader .)
ReadUInt16	(Inherited from BinaryReader .)
ReadUInt32	(Inherited from BinaryReader .)
ReadUInt64	(Inherited from BinaryReader .)
ReadVector2	Reads a Vector2 value.
ReadVector3	Reads a Vector3 value.
ReadVector4	Reads a Vector3 value.
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Dispose	(Inherited from BinaryReader .)
FillBuffer	(Inherited from BinaryReader .)
Finalize	(Inherited from Object .)
MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PacketReader Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketReader.ReadColor Method

Reads a Color value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Color ReadColor ()
```

Return Value

Color data received from the network packet.

See Also

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadDouble Method

Reads an 8-byte floating point value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override double ReadDouble ()
```

Return Value

8-byte floating point value from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadMatrix Method

Reads a [Matrix](#) value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Matrix ReadMatrix ()
```

Return Value

Matrix data from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadQuaternion Method

Reads a [Quaternion](#) value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Quaternion ReadQuaternion ()
```

Return Value

Quaternion data from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadSingle Method

Reads a 4-byte floating point value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override float ReadSingle ()
```

Return Value

Reads a 4-byte floating point value from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadVector2 Method

Reads a [Vector2](#) value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector2 ReadVector2 ()
```

Return Value

Vector data from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadVector3 Method

Reads a [Vector3](#) value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector3 ReadVector3 ()
```

Return Value

Vector data from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.ReadVector4 Method

Reads a [Vector3](#) value.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public Vector4 ReadVector4 ()
```

Return Value

Vector data from the received network packet.

See Also

Tasks

[How To: Receive Data](#)

Reference

[PacketReader Class](#)




[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader Properties

Public Properties

	Name	Description
	BaseStream	(Inherited from BinaryReader .)
	Length	Gets the length of the packet being read.
	Position	Gets or sets the current packet read position.

See Also

Reference

[PacketReader Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketReader.Length Property

Gets the length of the packet being read.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Length { get; }
```

Property Value

Length of the packet.

See Also

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketReader.Position Property

Gets or sets the current packet read position.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Position { get; set; }
```

Property Value

Current position of the packet.

See Also

Reference

[PacketReader Class](#)

[PacketReader Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter Class

Provides common functionality for efficiently formatting outgoing network packets.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class PacketWriter : BinaryWriter
```

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Members](#)


[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune




PacketWriter Members

The following tables list the members exposed by the PacketWriter type.










Public Constructors

	Name	Description
	PacketWriter	Overloaded. Initializes a new instance of PacketWriter.





Public Properties

	Name	Description
	BaseStream	(Inherited from BinaryWriter .)
	Length	Gets the length of the packet being written.
	Position	Gets or sets the current packet write position.

Public Methods

	Name	Description
	Close	(Inherited from BinaryWriter .)
	Equals	(Inherited from Object .)
	Flush	(Inherited from BinaryWriter .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Seek	(Inherited from BinaryWriter .)
	ToString	(Inherited from Object .)
	Write	Overloaded. Writes a value to an outgoing network packet.

Protected Methods

	Name	Description
	Dispose	(Inherited from BinaryWriter .)
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	Write7BitEncodedInt	(Inherited from BinaryWriter .)

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketWriter Fields

See Also

Reference

[PacketWriter Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketWriter Constructor

Initializes a new instance of **PacketWriter**.

Overload List

Name	Description
PacketWriter ()	Initializes an empty instance of PacketWriter .
PacketWriter (Int32)	Initializes a new instance of PacketWriter with the specified capacity.

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketWriter Constructor ()

Initializes an empty instance of **PacketWriter**.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PacketWriter ()
```

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter Constructor (Int32)

Initializes a new instance of **PacketWriter** with the specified capacity.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public PacketWriter (  
    int capacity  
)
```

Parameters

capacity

Capacity for the packet writer.

See Also

Reference

[PacketWriter Class](#)










[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)





Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter Methods

Public Methods

	Name	Description
	Close	(Inherited from BinaryWriter .)
	Equals	(Inherited from Object .)
	Flush	(Inherited from BinaryWriter .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Seek	(Inherited from BinaryWriter .)
	ToString	(Inherited from Object .)
	Write	Overloaded. Writes a value to an outgoing network packet.

Protected Methods

	Name	Description
	Dispose	(Inherited from BinaryWriter .)
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	Write7BitEncodedInt	(Inherited from BinaryWriter .)

See Also

Reference

[PacketWriter Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketWriter.Write Method

Writes a value to an outgoing network packet.

Overload List

Name	Description
PacketWriter.Write (Color)	Writes a Color value to an outgoing network packet.
PacketWriter.Write (Double)	Writes a Double value to an outgoing network packet.
PacketWriter.Write (Matrix)	Writes a Matrix value to an outgoing network packet.
PacketWriter.Write (Quaternion)	Writes a Quaternion value to an outgoing network packet.
PacketWriter.Write (Single)	Writes a Single value to an outgoing network packet.
PacketWriter.Write (Vector2)	Writes a Vector2 value to an outgoing network packet.
PacketWriter.Write (Vector3)	Writes a Vector3 value to an outgoing network packet.
PacketWriter.Write (Vector4)	Writes a Vector4 value to an outgoing network packet.
PacketWriter.Write (Boolean)	(Inherited from BinaryWriter .)
PacketWriter.Write (Byte)	(Inherited from BinaryWriter .)
PacketWriter.Write (Byte)	(Inherited from BinaryWriter .)
PacketWriter.Write (Byte, Int32, Int32)	(Inherited from BinaryWriter .)
PacketWriter.Write (Char)	(Inherited from BinaryWriter .)
PacketWriter.Write (Char)	(Inherited from BinaryWriter .)
PacketWriter.Write (Char, Int32, Int32)	(Inherited from BinaryWriter .)
PacketWriter.Write (Decimal)	(Inherited from BinaryWriter .)
PacketWriter.Write (Int16)	(Inherited from BinaryWriter .)
PacketWriter.Write (Int32)	(Inherited from BinaryWriter .)
PacketWriter.Write (Int64)	(Inherited from BinaryWriter .)
PacketWriter.Write (SByte)	(Inherited from BinaryWriter .)
PacketWriter.Write (String)	(Inherited from BinaryWriter .)
PacketWriter.Write (UInt16)	(Inherited from BinaryWriter .)
PacketWriter.Write (UInt32)	(Inherited from BinaryWriter .)
PacketWriter.Write (UInt64)	(Inherited from BinaryWriter .)

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketWriter.Write Method (Color)

Writes a [Color](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Write (  
    Color value  
)
```

Parameters

value

Value to be written.

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Double)

Writes a [Double](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override void Write (  
    double value  
)
```

Parameters

value

Value to be written.

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Matrix)

Writes a [Matrix](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Write (  
    Matrix value  
)
```

Parameters

value

Value being written.

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Quaternion)

Writes a [Quaternion](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Write (  
    Quaternion value  
)
```

Parameters

value

Value being written.

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Single)

Writes a [Single](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public override void Write (  
    float value  
)
```

Parameters

value

Value to be written.

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Vector2)

Writes a [Vector2](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Write (  
    Vector2 value  
)
```

Parameters

value

Value being written.

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Vector3)

Writes a [Vector3](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Write (  
    Vector3 value  
)
```

Parameters

value

Value being written.

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Write Method (Vector4)

Writes a [Vector4](#) value to an outgoing network packet.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Write (  
    Vector4 value  
)
```

Parameters

value

Value being written.

See Also

Tasks

[How To: Send Data](#)

Reference

[PacketWriter Class](#)




[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter Properties

Public Properties

	Name	Description
	BaseStream	(Inherited from BinaryWriter .)
	Length	Gets the length of the packet being written.
	Position	Gets or sets the current packet write position.

See Also

Reference

[PacketWriter Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

PacketWriter.Length Property

Gets the length of the packet being written.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Length { get; }
```

Property Value

Length of the packet.

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

PacketWriter.Position Property

Gets or sets the current packet write position.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int Position { get; set; }
```

Property Value

Current position of the packet.

See Also

Reference

[PacketWriter Class](#)

[PacketWriter Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

QualityOfService Class

Describes the quality of the network connection between this machine and the host of a multiplayer session that was discovered with a matchmaking query.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class QualityOfService
```

See Also

Reference

[QualityOfService Members](#)






[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune






QualityOfService Members

The following tables list the members exposed by the QualityOfService type.



Public Properties

Name	Description
 AverageRoundtripTime	Gets the average (median) round trip time of all the network packets that were sent during the quality of service measurement process.
 BytesPerSecondDownstream	Gets an estimate of the available downstream network bandwidth from the session host to this machine, measured in bytes per second.
 BytesPerSecondUpstream	Gets an estimate of the available upstream network bandwidth from this machine to the session host, measured in bytes per second.
 IsAvailable	Checks whether this quality of service operation has completed.
 MinimumRoundtripTime	Gets the minimum round trip time of any network packet that was sent during the quality of service measurement process.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[QualityOfService Class](#)



[Microsoft.Xna.Framework.Net Namespace](#)

QualityOfService Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[QualityOfService Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

QualityOfService Properties

Public Properties

	Name	Description
	AverageRoundtripTime	Gets the average (median) round trip time of all the network packets that were sent during the quality of service measurement process.
	BytesPerSecondDownstream	Gets an estimate of the available downstream network bandwidth from the session host to this machine, measured in bytes per second.
	BytesPerSecondUpstream	Gets an estimate of the available upstream network bandwidth from this machine to the session host, measured in bytes per second.
	IsAvailable	Checks whether this quality of service operation has completed.
	MinimumRoundtripTime	Gets the minimum round trip time of any network packet that was sent during the quality of service measurement process.

See Also

Reference

[QualityOfService Class](#)

[Microsoft.Xna.Framework.Net Namespace](#)

QualityOfService.AverageRoundtripTime Property

Gets the average (median) round trip time of all the network packets that were sent during the quality of service measurement process.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan AverageRoundtripTime { get; }
```

Property Value

The average (median) round trip time of all the network packets that were sent during the quality of service measurement process. Will return zero if the query has not yet completed. You can check if the query is complete using [IsAvailable](#).

See Also

Reference

[QualityOfService Class](#)

[QualityOfService Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

QualityOfService.BytesPerSecondDownstream Property

Gets an estimate of the available downstream network bandwidth from the session host to this machine, measured in bytes per second.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BytesPerSecondDownstream { get; }
```

Property Value

An estimate of the available downstream network bandwidth from the session host to this machine, measured in bytes per second. Will return zero if the query has not yet completed. You can check if the query is complete using [IsAvailable](#).

Remarks

The bandwidth algorithm used for [BytesPerSecondUpstream](#) and [BytesPerSecondDownstream](#) gives only an estimate of bandwidth and there is a possibility it may not accurately reflect the true bandwidth provided by the player's internet service provider. To avoid inaccurately representing the true bandwidth to the player, games should avoid displaying [BytesPerSecondUpstream](#) and [BytesPerSecondDownstream](#).

Even though the results of these properties provide only rough estimates, the values can be useful for relative comparisons or host stack ranking. For instance, these properties might be used for choosing a host gaming machine with a bandwidth twice as large as another gaming machine or sorting a list based on bandwidth.

See Also

Reference

[QualityOfService Class](#)

[QualityOfService Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

QualityOfService.BytesPerSecondUpstream Property

Gets an estimate of the available upstream network bandwidth from this machine to the session host, measured in bytes per second.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public int BytesPerSecondUpstream { get; }
```

Property Value

An estimate of the available upstream network bandwidth from this machine to the session host, measured in bytes per second. Will return zero if the query has not yet completed. You can check if the query is complete using [IsAvailable](#).

Remarks

The bandwidth algorithm used for [BytesPerSecondUpstream](#) and [BytesPerSecondDownstream](#) gives only an estimate of bandwidth and there is a possibility it may not accurately reflect the true bandwidth provided by the player's internet service provider. To avoid inaccurately representing the true bandwidth to the player, games should avoid displaying [BytesPerSecondUpstream](#) and [BytesPerSecondDownstream](#).

Even though the results of these properties provide only rough estimates, the values can be useful for relative comparisons or host stack ranking. For instance, these properties might be used for choosing a host gaming machine with a bandwidth twice as large as another gaming machine or sorting a list based on bandwidth.

See Also

Reference

[QualityOfService Class](#)

[QualityOfService Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

QualityOfService.IsAvailable Property

Checks whether this quality of service operation has completed.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsAvailable { get; }
```

Property Value

true if quality of service data is available; **false** otherwise.

Remarks Matchmaking search results are returned before the quality of service probing has finished, so as to avoid an excessive delay before you can display the list of sessions to the user, but the quality of service checks continue in the background even after the search results are available. Initially the **IsAvailable** property will be set to **false**, and all the other properties will be zero, but after the probing finishes this will become true and the other properties will be filled in with some actual data.

See Also

Reference

[QualityOfService Class](#)

[QualityOfService Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

QualityOfService.MinimumRoundtripTime Property

Gets the minimum round trip time of any network packet that was sent during the quality of service measurement process.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public TimeSpan MinimumRoundtripTime { get; }
```

Property Value

The minimum round trip time of any network packet that was sent during the quality of service measurement process. Will return zero if the query has not yet completed. You can check if the query is complete using [IsAvailable](#).

See Also

Reference

[QualityOfService Class](#)

[QualityOfService Members](#)

[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

SendDataOptions Enumeration

Defines options for network packet transmission.

Namespace: Microsoft.Xna.Framework.Net

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
[FlagsAttribute]
public enum SendDataOptions
```

Members

Member name	Description
None	<p>Sends the data with no guarantees.</p> <p>Packets of this type may be delivered in any order, with occasional packet loss.</p> <p>This is the most efficient option in terms of network bandwidth and machine resource usage. However, it is recommended only in situations where your game can recover from occasional packet loss.</p>
Reliable	<p>Sends the data with reliable delivery, but no special ordering.</p> <p>Packets of this type are resent until arrival at the destination. They may arrive out of order.</p>
InOrder	<p>Sends the data with guaranteed ordering, but without reliable delivery.</p> <p>Occasionally, packets of this type are not delivered. However, any delivered packets always arrive in the order in which they are sent.</p> <p>Use this option in situations where the transmitted value changes constantly. Old versions never arrive after a more recent version.</p>
ReliableInOrder	<p>Sends the data with reliability and arrival in the order originally sent.</p> <p>Packets of this type are resent until arrival and ordered internally. This means they arrive in the same order in which they were sent.</p> <p>In terms of network bandwidth usage, this is the strongest and most expensive option. Use this only when arrival and ordering are essential. Commonly, a game uses this option for a small percentage of packets. The majority of gameplay data is sent using <i>None</i> or <i>Reliable</i>.</p>
Chat	<p>Mark that this packet contains chat data, such as a player-to-player message string entered using the keyboard. To comply with international regulations, you must send such data without packet encryption. Therefore, you must use this flag to mark it. To maintain security, other game data should not use this flag. It is acceptable and efficient to mix encrypted and unencrypted data.</p> <p>If you send packets both with and without this flag within a single frame, both the encrypted and unencrypted data streams will be merged into a single wire packet. This option can be combined with either or both of the <i>Reliable</i> and <i>InOrder</i> flags. When you request in-order delivery for chat packets, they will be ordered relative to other chat packets, but they may arrive out of order with respect to other non-chat data.</p>

See Also

Reference




[Microsoft.Xna.Framework.Net Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

Microsoft.Xna.Framework.Storage Namespace

Contains classes that allow reading and writing of files.

Classes

	Name	Description
	StorageContainer	Represents a logical collection of storage files.
	StorageDevice	Represents a storage device for user data, such as a memory unit or hard drive.
	StorageDeviceNotConnectedException	The exception that is thrown when the requested StorageDevice is not connected.

See Also

Tasks

[How To: Create a File](#)

[How To: Open a File](#)

[How To: Copy a File](#)

[How To: Rename a File](#)

[How To: Enumerate Files](#)

[How To: Delete a File](#)

[How To: Load a Game Data File](#)

[How To: Serialize Data](#)

[How To: Get a StorageDevice Asynchronously](#)

Concepts

[Storage Overview](#)

StorageContainer Class

Represents a logical collection of storage files.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class StorageContainer : IDisposable
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageContainer Members](#)






[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune







StorageContainer Members

The following tables list the members exposed by the StorageContainer type.



Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Path	Gets the file path to the location of the user's save game files.
	StorageDevice	Gets the StorageDevice that holds the files in this container.
	TitleLocation	Gets the file path to the location of the title storage space.
	TitleName	Gets the name of the title.


Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Concepts

[Storage Overview](#)







Reference

[StorageContainer Class](#)



[Microsoft.Xna.Framework.Storage Namespace](#)

StorageContainer Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[StorageContainer Class](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageContainer.Dispose Method

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public void Dispose ()
```

Remarks

After the [StorageContainer](#) has been opened on a [StorageDevice](#), changes to the files within the container are cached until the container is disposed. If the storage device is removed or the Xbox 360 is powered down before the container is disposed, any changes are lost.

See Also

Reference

[StorageContainer Class](#)

[StorageContainer Members](#)

[OpenContainer](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer.Finalize Method

Allows this object to attempt to free resources and perform other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected override void Finalize ()
```

Remarks

This method overrides [System.Object.Finalize](#). Application code should not call this method. An object's [Finalize](#) method is automatically invoked during garbage collection, unless a call to the [GC.SuppressFinalize](#) method has disabled finalization by the garbage collector.

See Also

Reference

[StorageContainer Class](#)






[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer Properties

Public Properties

	Name	Description
	IsDisposed	Gets a value that indicates whether the object is disposed.
	Path	Gets the file path to the location of the user's save game files.
	StorageDevice	Gets the StorageDevice that holds the files in this container.
 S	TitleLocation	Gets the file path to the location of the title storage space.
	TitleName	Gets the name of the title.

See Also

Reference

[StorageContainer Class](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageContainer.IsDisposed Property

Gets a value that indicates whether the object is disposed.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

true if the object is disposed; **false** otherwise.

See Also

Reference

[StorageContainer Class](#)

[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer.Path Property

Gets the file path to the location of the user's save game files.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string Path { get; }
```

Property Value

The file path to the location of the user's save game files.

See Also

Reference

[StorageContainer Class](#)

[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer.StorageDevice Property

Gets the [StorageDevice](#) that holds the files in this container.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StorageDevice StorageDevice { get; }
```

Property Value

The [StorageDevice](#) that holds the files in this container.

See Also

Reference

[StorageContainer Class](#)

[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer.TitleLocation Property

Gets the file path to the location of the title storage space.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static string TitleLocation { get; }
```

Property Value

The title's install location based on the current platform.

Remarks

Use this member to create full paths to files in the same location as the game executable. For example:

C#

```
using System;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Storage;
using System.IO;
using System.Xml.Serialization;
using System.Diagnostics;
/// <summary>
/// This method opens a file using System.IO classes and the
/// TitleLocation property. It presumes that a file named
/// ship.dds has been deployed alongside the game.
/// </summary>
private static void DoOpenFile()
{
    FileStream file = OpenTitleFile(
        "ship.dds", FileMode.Open, FileAccess.Read);
    Console.WriteLine("File Size: " + file.Length);
    file.Close();
}
private static FileStream OpenTitleFile(
    string filename, FileMode mode, FileAccess access)
{
    string fullpath = Path.Combine(StorageContainer.TitleLocation, filename);
    return File.Open(fullpath, mode, access);
}
```

See Also

Reference

[StorageContainer Class](#)

[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer.TitleName Property

Gets the name of the title.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public string TitleName { get; }
```

Property Value

The name of the title.

Remarks

This is the same as the *titleName* passed to [OpenContainer](#).

See Also

Reference

[StorageContainer Class](#)


[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageContainer Events

Public Events

	Name	Description
	Disposing	Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

See Also

Reference

[StorageContainer Class](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageContainer.Disposing Event

Occurs when Dispose is called or when this object is finalized and collected by the garbage collector of the Microsoft .NET common language runtime.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public event EventHandler Disposing
```

See Also

Reference

[StorageContainer Class](#)

[StorageContainer Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDevice Class

Represents a storage device for user data, such as a memory unit or hard drive.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public sealed class StorageDevice
```

See Also

Concepts

[Storage Overview](#)

Reference

[StorageDevice Members](#)




[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune








StorageDevice Members

The following tables list the members exposed by the StorageDevice type.



Public Properties

	Name	Description
	FreeSpace	Gets the amount of free space on the device.
	IsConnected	Gets whether the device is connected.
	TotalSpace	Gets the total amount of space on the device.


Public Methods

	Name	Description
	DeleteContainer	Deletes a storage container without opening it.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	OpenContainer	Opens a StorageContainer containing any files for the specified title.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Public Events

	Name	Description
	S DeviceChanged	Occurs when a device is removed or inserted.

See Also

Concepts

[Storage Overview](#)








Reference

[StorageDevice Class](#)



[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDevice Methods

Public Methods

	Name	Description
	DeleteContainer	Deletes a storage container without opening it.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	OpenContainer	Opens a StorageContainer containing any files for the specified title.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[StorageDevice Class](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDevice.DeleteContainer Method

Deletes a storage container without opening it. You can do this even if the container is corrupted.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#


```
public void DeleteContainer (
    string titleName
)
```

Parameters

titleName

The name of the storage container to delete.

Exceptions

Exception type	Condition
ArgumentNullException	<i>titleName</i> is null .
InvalidOperationException	The container is currently open. A container cannot be deleted if it is open.  Note No exception will be thrown if the container does not exist.

Remarks

The string passed to *titleName* must be the same string used with [StorageDevice.OpenContainer](#).

Usually a storage container is closed with [StorageContainer.Dispose](#) so that it can be re-opened with its data intact. Use [DeleteContainer](#) only when you want to delete the data permanently or when you determine the storage container has been corrupted.

See the [Storage Overview](#) for more information about using storage containers in your title.

See Also

Reference

[StorageDevice Class](#)

[StorageDevice Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDevice.OpenContainer Method

Opens a [StorageContainer](#) containing any files for the specified title.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StorageContainer OpenContainer (
    string titleName
)
```

Parameters

titleName

The name of the XNA Framework title.

Return Value

A [StorageContainer](#) containing any files for the title.

Exceptions

Exception type	Condition
ArgumentNullException	<i>titleName</i> is null .
InvalidOperationException	A container is already open for the PlayerIndex set in BeginShowStorageDeviceSelector . A new container cannot be opened until all previous containers used by this PlayerIndex have been disposed.

Remarks

titleName should be a constant human readable string used to identify your title and generate a title-unique storage location.

After the container has been opened, changes to the files within the container are cached until the container is disposed. For this reason, only one container per profile can be opened at the same time on the same device.

If the storage device is removed or the Xbox 360 console is powered down before the container is disposed, any changes are lost.

Call the [Dispose](#) method to dispose a [StorageContainer](#).

See Also

Reference

[StorageDevice Class](#)

[StorageDevice Members](#)




[Dispose](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDevice Properties

Public Properties

	Name	Description
	FreeSpace	Gets the amount of free space on the device.
	IsConnected	Gets whether the device is connected.
	TotalSpace	Gets the total amount of space on the device.

See Also

Reference

[StorageDevice Class](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDevice.FreeSpace Property

Gets the amount of free space on the device.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public long FreeSpace { get; }
```

Property Value

Free space on the device, in bytes.

Exceptions

Exception type	Condition
StorageDeviceNotConnectedException	The storage device bound to the container is not connected.

See Also

Reference

[StorageDevice Class](#)

[StorageDevice Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDevice.IsConnected Property

Gets whether the device is connected.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

true if the device is connected; **false** otherwise.

See Also

Reference

[StorageDevice Class](#)

[StorageDevice Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDevice.TotalSpace Property

Gets the total amount of space on the device.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public long TotalSpace { get; }
```

Property Value

Total amount of space on the device, in bytes.

Exceptions

Exception type	Condition
StorageDeviceNotConnectedException	The storage device bound to the container is not connected.

See Also

Reference

[StorageDevice Class](#)


[StorageDevice Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDevice Events

Public Events

	Name	Description
 S	DeviceChanged	Occurs when a device is removed or inserted.

See Also

Reference

[StorageDevice Class](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDevice.DeviceChanged Event

Occurs when a device is removed or inserted.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public static event EventHandler<EventArgs> DeviceChanged
```

See Also

Reference

[StorageDevice Class](#)

[StorageDevice Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDeviceNotConnectedException Class

The exception that is thrown when the requested [StorageDevice](#) is not connected.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public class StorageDeviceNotConnectedException : ExternalException
```

See Also

Reference

[StorageDeviceNotConnectedException Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune









StorageDeviceNotConnectedException Members

The following tables list the members exposed by the StorageDeviceNotConnectedException type.


Public Constructors

Name	Description
 StorageDeviceNotConnectedException	Overloaded. Initializes a new instance of this class.







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 ErrorCode	(Inherited from ExternalException .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[StorageDeviceNotConnectedException Class](#)
[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDeviceNotConnectedException Constructor

Initializes a new instance of this class.

Overload List

Name	Description
StorageDeviceNotConnectedException ()	Initializes a new instance of this class.
StorageDeviceNotConnectedException (SerializationInfo, StreamingContext)	Initializes a new instance of this class during deserialization.
StorageDeviceNotConnectedException (String)	Initializes a new instance of this class with a specified error message.
StorageDeviceNotConnectedException (String, Exception)	Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[StorageDeviceNotConnectedException Class](#)

[StorageDeviceNotConnectedException Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDeviceNotConnectedException Constructor ()

Initializes a new instance of this class.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StorageDeviceNotConnectedException ()
```

Remarks

This constructor initializes the [Message](#) property of the new instance to a system-supplied message that describes the error, such as "A file read or write operation failed." This message takes into account the current system culture.

The following table shows the initial property values for an instance of [StorageDeviceNotConnectedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[StorageDeviceNotConnectedException Class](#)

[StorageDeviceNotConnectedException Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDeviceNotConnectedException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of this class during deserialization.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
protected StorageDeviceNotConnectedException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

The information needed to serialize an object.

context

The source or destination for the serialization stream.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

See Also

Reference

[StorageDeviceNotConnectedException Class](#)

[StorageDeviceNotConnectedException Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

PlatformsWindows XP SP2, Windows Vista

StorageDeviceNotConnectedException Constructor (String)

Initializes a new instance of this class with a specified error message.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StorageDeviceNotConnectedException (  
    string message  
)
```

Parameters

message

A message that describes the error.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

The following table shows the initial property values for an instance of [StorageDeviceNotConnectedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[StorageDeviceNotConnectedException Class](#)

[StorageDeviceNotConnectedException Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)

Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDeviceNotConnectedException Constructor (String, Exception)

Initializes a new instance of this class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Storage

Assembly: Microsoft.Xna.Framework (in microsoft.xna.framework.dll)

Syntax

C#

```
public StorageDeviceNotConnectedException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The exception that is the cause of the current exception. If the *innerException* parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Remarks

The content of the message parameter is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

An exception that is thrown as a direct result of a previous exception should include a reference to the previous exception in the [InnerException](#) property. The [InnerException](#) property returns the same value that is passed into the constructor, or a null reference if the [InnerException](#) property does not supply the inner exception value to the constructor.

The following table shows the initial property values for an instance of [StorageDeviceNotConnectedException](#).

Property	Value
InnerException	A null reference.
Message	The localized error message string.

See Also

Reference

[StorageDeviceNotConnectedException Class](#)







[StorageDeviceNotConnectedException Members](#)

[Microsoft.Xna.Framework.Storage Namespace](#)



Platforms Xbox 360, Windows XP SP2, Windows Vista, Zune

StorageDeviceNotConnectedException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)









See Also

Reference


[StorageDeviceNotConnectedException Class](#)
[Microsoft.Xna.Framework.Storage Namespace](#)

StorageDeviceNotConnectedException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	ErrorCode	(Inherited from ExternalException .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[StorageDeviceNotConnectedException Class](#)
[Microsoft.Xna.Framework.Storage Namespace](#)

Content Pipeline Class Library

The Content Pipeline class library is a library of classes, interfaces, and value types that are included in XNA Game Studio. This library provides access to XNA Framework Content Pipeline functionality and is designed to be the foundation on which Content Pipeline–related applications, components, and controls are built.

Namespaces

[Microsoft.Xna.Framework.Content.Pipeline](#)

Provides classes representing base types and building block functionality for use by more specialized object models, such as the Graphics DOM.

[Microsoft.Xna.Framework.Content.Pipeline.Audio](#)

Provides intermediate classes and types for representing and manipulating graphics audio data.

[Microsoft.Xna.Framework.Content.Pipeline.Graphics](#)

Provides intermediate classes and types for representing and manipulating graphics data.

[Microsoft.Xna.Framework.Content.Pipeline.Processors](#)

Provides base classes that represent processors used by the XNA Framework Content Pipeline when processing specific game asset types.

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler](#)

Provides base classes that represent compilers and writers used by the XNA Framework Content Pipeline when processing specific game asset types.

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate](#)

Provides base classes that represent the creation and writing of intermediate content for game asset types processed by the XNA Framework Content Pipeline.

[Microsoft.Xna.Framework.Content.Pipeline.Tasks](#)

Provides support for importing and processing game assets into the binary format that is used by the content loader of a game project.

See Also

Concepts

[Content Pipeline](#)

[Overview of the Content Pipeline](#)

[Content Pipeline Content Catalog at XNA Creators Club Online](#)

Microsoft.Xna.Framework.Content.Pipeline Namespace

Note

This namespace is available only when developing for Windows.



Provides classes representing base types and building block functionality for use by more specialized object models, such as the Graphics DOM.

Classes

Name	Description
ChildCollection	Provides a collection of child objects for a content item.
ContentBuildLogger	Provides methods for reporting informational messages or warnings from content importers and processors.
ContentIdentity	Provides properties describing the origin of the game asset, such as the original source file and creation tool. This information is used for error reporting, and by processors that need to determine from what directory the asset was originally loaded.
ContentImporter	Implements a file format importer for use with game assets.
ContentImporterAttribute	Provides properties that identify and provide metadata about the importer, such as supported file extensions and caching information.
ContentImporterContext	Provides properties that define logging behavior for the importer.
ContentItem	Provides properties that define various aspects of content stored using the intermediate file format of the XNA Framework.
ContentProcessor	Provides a base class to use when developing custom processor components. All processors must derive from this class.
ContentProcessorAttribute	Gets any existing content processor components.
ContentProcessorContext	Provides access to custom processor parameters, methods for converting member data, and triggering nested builds.
ExternalReference	Specifies external references to a data file for the content item.
FontDescriptionImporter	Provides methods for reading .spritefont files for use in the Content Pipeline.
InvalidContentException	Thrown when errors are encountered in content during processing.
NamedValueDictionary	Base class for dictionaries that map string identifiers to data values.
OpaqueDataDictionary	Provides properties that define opaque data for a game asset.
PipelineComponentScanner	Implements a scanner object containing the available importers and processors for an application.
PipelineException	Thrown when errors are encountered during a content pipeline build.
ProcessorParameter	Represents a processor parameter.
ProcessorParameterCollection	Represents a collection of processor parameters , usually for a single processor.
VideoContent	Provides a base class for all video objects.
XmlImporter	Implements an importer for reading intermediate XML files.

Interfaces

Name	Description
------	-------------

 IContentImporter	Accesses a statically typed ContentImporter instance from generic code using dynamic typing.
 IContentProcessor	Provides methods and properties for accessing a statically typed ContentProcessor subclass, using dynamically typed object data.

ChildCollection Generic Class

Note

This generic class is available only when developing for Windows.

Provides a collection of child objects for a content item.

Links from a child object to its parent are maintained as the collection contents are modified.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ChildCollection<TParent,TChild> : Collection<TChild>
```

See Also

Reference

[ChildCollection Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista



ChildCollection Members

The following tables list the members exposed by the ChildCollection type.

Protected Constructors

Name	Description
 ChildCollection	Creates an instance of ChildCollection .

















Public Properties

Name	Description
 Count	(Inherited from Collection .)
 Item	(Inherited from Collection .)







Protected Properties

Name	Description
 Items	(Inherited from Collection .)

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 SetItem	Overloaded. Modifies the value of a child object.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	Removes all children from the collection.
 Finalize	(Inherited from Object .)
 GetParent	Gets the parent of a child object.
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes a child object from the collection.
 SetParent	Modifies the value of the parent object of the specified child object.

See Also

Reference

[ChildCollection Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ChildCollection Constructor

Note

This constructor is available only when developing for Windows.

Creates an instance of [ChildCollection](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ChildCollection (  
    TParent parent  
)
```

Parameters

parent

Parent object of the child objects returned in the collection.

See Also

Reference

[ChildCollection Generic Class](#)

















[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)







PlatformsWindows XP SP2, Windows Vista

ChildCollection Methods

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 SetItem	Overloaded. Modifies the value of a child object.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	Removes all children from the collection.
 Finalize	(Inherited from Object .)
 GetParent	Gets the parent of a child object.
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes a child object from the collection.
 SetParent	Modifies the value of the parent object of the specified child object.

See Also

Reference

[ChildCollection Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ChildCollection.ClearItems Method

Note

This method is available only when developing for Windows.

Removes all children from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void ClearItems ()
```

See Also

Reference

[ChildCollection Generic Class](#)

[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ChildCollection.GetParent Method

Note

This method is available only when developing for Windows.

Gets the parent of a child object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected abstract TParent GetParent (  
    TChild child  
)
```

Parameters

child

The child of the parent being retrieved.

Return Value

The parent of the child object.

See Also

Reference

[ChildCollection Generic Class](#)

[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ChildCollection.InsertItem Method

Inserts a child object into the collection at the specified location.

Overload List

Name	Description
ChildCollection.InsertItem (Int32, TChild)	Inserts a child object into the collection at the specified location.
ChildCollection.InsertItem (Int32, TChild)	(Inherited from Collection .)

See Also

Reference

[ChildCollection](#) Generic Class

[ChildCollection](#) Members

[Microsoft.Xna.Framework.Content.Pipeline](#) Namespace

ChildCollection.InsertItem Method (Int32, TChild)

Note

This method is available only when developing for Windows.

Inserts a child object into the collection at the specified location.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void InsertItem (  
    int index,  
    TChild item  
)
```

Parameters

index

The position in the collection.

item

The child object being inserted.

See Also

Reference

[ChildCollection Generic Class](#)

[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ChildCollection.RemoveItem Method

Note

This method is available only when developing for Windows.

Removes a child object from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void RemoveItem (  
    int index  
)
```

Parameters

index

The index of the item being removed.

See Also

Reference

[ChildCollection Generic Class](#)

[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ChildCollection.SetItem Method

Modifies the value of a child object.

Overload List

Name	Description
ChildCollection.SetItem (Int32, TChild)	Modifies the value of the child object at the specified location.
ChildCollection.SetItem (Int32, TChild)	(Inherited from Collection .)

See Also

Reference

[ChildCollection](#) Generic Class

[ChildCollection](#) Members

[Microsoft.Xna.Framework.Content.Pipeline](#) Namespace

ChildCollection.SetItem Method (Int32, TChild)

Note

This method is available only when developing for Windows.

Modifies the value of the child object at the specified location.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void SetItem (  
    int index,  
    TChild item  
)
```

Parameters

index

The index of the child object being modified.

item

The new value for the child object.

See Also

Reference

[ChildCollection Generic Class](#)

[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ChildCollection.SetParent Method

Note

This method is available only when developing for Windows.

Modifies the value of the parent object of the specified child object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected abstract void SetParent (  
    TChild child,  
    TParent parent  
)
```

Parameters

child

The child of the parent being modified.

parent

The new value for the parent object.

See Also

Reference

[ChildCollection Generic Class](#)



[ChildCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ChildCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[ChildCollection Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentBuildLogger Class

Note

This class is available only when developing for Windows.

Provides methods for reporting informational messages or warnings from content importers and processors.

Do not use this class to report errors. Instead, report errors by throwing a [PipelineException](#) or [InvalidContentException](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentBuildLogger
```

See Also

Reference

[ContentBuildLogger Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista


ContentBuildLogger Members

The following tables list the members exposed by the ContentBuildLogger type.











Protected Constructors

Name	Description
 ContentBuildLogger	Initializes a new instance of ContentBuildLogger.




Public Properties

Name	Description
 LoggerRootDirectory	Gets or sets the base reference path used when reporting errors during the content build process.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 LogImportantMessage	Outputs a high-priority status message from a content importer or processor.
 LogMessage	Outputs a low priority status message from a content importer or processor.
 LogWarning	Outputs a warning message from a content importer or processor.
 PopFile	Outputs a message indicating that a content asset has completed processing.
 PushFile	Outputs a message indicating that a content asset has begun processing.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetCurrentFilename	Gets the filename currently being processed, for use in warning and error messages.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentBuildLogger Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentBuildLogger Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **ContentBuildLogger**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentBuildLogger ()
```

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentBuildLogger Methods

Public Methods

Name	Description
Equals	(Inherited from Object .)
GetHashCode	(Inherited from Object .)
GetType	(Inherited from Object .)
LogImportantMessage	Outputs a high-priority status message from a content importer or processor.
LogMessage	Outputs a low priority status message from a content importer or processor.
LogWarning	Outputs a warning message from a content importer or processor.
PopFile	Outputs a message indicating that a content asset has completed processing.
PushFile	Outputs a message indicating that a content asset has begun processing.
ReferenceEquals	(Inherited from Object .)
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
GetCurrentFilename	Gets the filename currently being processed, for use in warning and error messages.
MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentBuildLogger Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentBuildLogger.GetCurrentFilename Method

Note

This method is available only when developing for Windows.

Gets the filename currently being processed, for use in warning and error messages.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected string GetCurrentFilename (  
    ContentIdentity contentIdentity  
)
```

Parameters

contentIdentity

Identity of a content item. If specified, **GetCurrentFilename** uses this value to refine the search. If no value is specified, the current [PushFile](#) state is used.

Return Value

Name of the file being processed.

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentBuildLogger.LogImportantMessage Method

Note

This method is available only when developing for Windows.

Outputs a high-priority status message from a content importer or processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract void LogImportantMessage (
    string message,
    Object[] messageArgs
)
```

Parameters

message

Message being reported.

messageArgs

[[ParamArrayAttribute](#)] Arguments for the reported message.

Remarks

Use **LogImportantMessage** only for important information, because these messages appear in the **Output** window of Visual Studio. Importers and processors should use this functionality only for unusual or unexpected situations.

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentBuildLogger.LogMessage Method

Note

This method is available only when developing for Windows.

Outputs a low priority status message from a content importer or processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract void LogMessage (
    string message,
    Object[] messageArgs
)
```

Parameters

message

Message being reported.

messageArgs

[[ParamArrayAttribute](#)] Arguments for the reported message.

Remarks

This message will not usually appear in the **Build Output** window of Visual Studio, but it can be viewed if the content pipeline is called directly from an MSBuild task.

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentBuildLogger.LogWarning Method

Note

This method is available only when developing for Windows.

Outputs a warning message from a content importer or processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract void LogWarning (
    string helpLink,
    ContentIdentity contentIdentity,
    string message,
    Object[] messageArgs
)
```

Parameters

helpLink

Link to an existing online help topic containing related information.

contentIdentity

Identity of the content item that generated the message.

message

Message being reported.

messageArgs

[[ParamArrayAttribute](#)] Arguments for the reported message.

Remarks

Use **LogWarning** only for important information, because these messages appear in the **Error List** window of Visual Studio. The message text should indicate what steps need to be taken to correct the problem.

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentBuildLogger.PopFile Method

Note

This method is available only when developing for Windows.

Outputs a message indicating that a content asset has completed processing.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void PopFile ()
```

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentBuildLogger.PushFile Method

Note

This method is available only when developing for Windows.

Outputs a message indicating that a content asset has begun processing.

All logger warnings or error exceptions from this time forward to the next [PopFile](#) call refer to this file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void PushFile (  
    string filename  
)
```

Parameters

filename

Name of the file containing future messages.

See Also

Reference

[ContentBuildLogger Class](#)


[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentBuildLogger Properties

Public Properties

	Name	Description
	LoggerRootDirectory	Gets or sets the base reference path used when reporting errors during the content build process.

See Also

Reference

[ContentBuildLogger Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentBuildLogger.LoggerRootDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the base reference path used when reporting errors during the content build process.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string LoggerRootDirectory { get; set; }
```

Property Value

Current name of the base directory or the value to be set.

Remarks

When building XNA content, all errors are displayed in the Error List Window of Visual Studio. Along with the data that triggered the error, the Error List displays the path and filename where the offending data was encountered.

By default, the path of the offending file that is reported in the Error List is always expressed as it relates to the location of the project file that references it.

When set, the **LoggerRootDirectory** property specifies an explicit base path for error reporting that overrides the default operation. All offending files reported in the Error List will be expressed as they relate to the base path set in **LoggerRootDirectory**.

See Also

Reference

[ContentBuildLogger Class](#)

[ContentBuildLogger Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentIdentity Class

Note

This class is available only when developing for Windows.

Provides properties describing the origin of the game asset, such as the original source file and creation tool. This information is used for error reporting, and by processors that need to determine from what directory the asset was originally loaded.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[SerializableAttribute]  
public class ContentIdentity
```

See Also

Reference

[ContentIdentity Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista




ContentIdentity Members

The following tables list the members exposed by the ContentIdentity type.






Public Constructors

	Name	Description
	ContentIdentity	Overloaded. Initializes a new instance of ContentIdentity .



Public Properties

	Name	Description
	FragmentIdentifier	Gets or sets the specific location of the content item within the larger source file.
	SourceFilename	Gets or sets the file name of the asset source.
	SourceTool	Gets or sets the creation tool of the asset.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentIdentity Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentIdentity Constructor

Initializes a new instance of [ContentIdentity](#).

Overload List

Name	Description
ContentIdentity ()	Initializes a new instance of ContentIdentity .
ContentIdentity (String)	Initializes a new instance of ContentIdentity with the specified values.
ContentIdentity (String, String)	Initializes a new instance of ContentIdentity with the specified values.
ContentIdentity (String, String, String)	Initializes a new instance of ContentIdentity with the specified values.

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentIdentity Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentIdentity](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentIdentity ()
```

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentIdentity Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentIdentity](#) with the specified values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentIdentity (  
    string sourceFilename  
)
```

Parameters

sourceFilename

The absolute path to the file name of the asset source.

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentIdentity Constructor (String, String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentIdentity](#) with the specified values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentIdentity (  
    string sourceFilename,  
    string sourceTool  
)
```

Parameters

sourceFilename

The absolute path to the file name of the asset source.

sourceTool

The name of the digital content creation (DCC) tool that created the asset.

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentIdentity Constructor (String, String, String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentIdentity](#) with the specified values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentIdentity (  
    string sourceFilename,  
    string sourceTool,  
    string fragmentIdentifier  
)
```

Parameters

sourceFilename

The absolute path to the file name of the asset source.

sourceTool

The name of the digital content creation (DCC) tool that created the asset.

fragmentIdentifier

Specific location of the content item within the larger source file. For example, this could be a line number in the file.

See Also

Reference

[ContentIdentity Class](#)






[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentIdentity Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also




Reference

[ContentIdentity Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentIdentity Properties

Public Properties

	Name	Description
	FragmentIdentifier	Gets or sets the specific location of the content item within the larger source file.
	SourceFilename	Gets or sets the file name of the asset source.
	SourceTool	Gets or sets the creation tool of the asset.

See Also

Reference

[ContentIdentity Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentIdentity.FragmentIdentifier Property

Note

This property is available only when developing for Windows.

Gets or sets the specific location of the content item within the larger source file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string FragmentIdentifier { get; set; }
```

Property Value

Location of the content item. This location can be in various forms. For example, this could indicate a line number within the source file.

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentIdentity.SourceFilename Property

Note

This property is available only when developing for Windows.

Gets or sets the file name of the asset source.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string SourceFilename { get; set; }
```

Property Value

The absolute path to the file name of the asset source.

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentIdentity.SourceTool Property

Note

This property is available only when developing for Windows.

Gets or sets the creation tool of the asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string SourceTool { get; set; }
```

Property Value

The name of the digital content creation (DCC) tool that created the asset.

See Also

Reference

[ContentIdentity Class](#)

[ContentIdentity Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporter Generic Class

Note

This generic class is available only when developing for Windows.

Implements a file format importer for use with game assets.

Importers, either provided by the framework or written by a developer, must derive from **ContentImporter**, as well as being marked with a [ContentImporterAttribute](#).

An importer should produce results in the standard intermediate object model. If an asset has information not supported by the object model, the importer should output it as opaque data (key/value attributes attached to the relevant object). By following this procedure, a content pipeline can access specialized digital content creation (DCC) tool information, even when that information has not been fully standardized into the official object model.

You can also design custom importers that accept and import types containing specific third-party extensions to the object model.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentImporter<T> : IContentImporter
```

See Also

Reference

[ContentImporter Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista







ContentImporter Members

The following tables list the members exposed by the ContentImporter type.



Protected Constructors

	Name	Description
	ContentImporter	Initializes a new instance of ContentImporter.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Import	Called by the framework when importing a game asset.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Content.Pipeline.IContentImporter.Import	Called by the framework when importing a game asset.

See Also

Reference

[ContentImporter Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **ContentImporter**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentImporter ()
```

See Also

Reference

[ContentImporter Generic Class](#)







[ContentImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentImporter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Import	Called by the framework when importing a game asset.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Content.Pipeline.IContentImporter.Import	Called by the framework when importing a game asset.

See Also

Reference

[ContentImporter Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporter.Import Method

Note

This method is available only when developing for Windows.

Called by the framework when importing a game asset. This is the method called by XNA when an asset is to be imported into an object that can be recognized by the Content Pipeline.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract T Import (  
    string filename,  
    ContentImporterContext context  
)
```

Parameters

filename

Name of a game asset file.

context

Contains information for importing a game asset, such as a logger interface.

Return Value

Resulting game asset.

See Also

Reference

[ContentImporter Generic Class](#)

[ContentImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Microsoft.Xna.Framework.Content.Pipeline.IContentImporter.Import Method

Note

This method is available only when developing for Windows.

Called by the framework when importing a game asset. This is the method called by XNA when an asset is to be imported into an object that can be recognized by the Content Pipeline.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private Object Microsoft.Xna.Framework.Content.Pipeline.IContentImporter.Import (  
    string filename,  
    ContentImporterContext context  
)
```

Parameters

filename

Name of a game asset file.

context

Contains information for importing a game asset, such as a logger interface.

Return Value

Resulting game asset.

See Also

Reference

[ContentImporter Generic Class](#)

[ContentImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterAttribute Class

Note

This class is available only when developing for Windows.

Provides properties that identify and provide metadata about the importer, such as supported file extensions and caching information.

Importers are required to initialize this attribute.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class ContentImporterAttribute : Attribute
```

See Also

Reference

[ContentImporterAttribute Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista






ContentImporterAttribute Members

The following tables list the members exposed by the ContentImporterAttribute type.










Public Constructors

Name	Description
 ContentImporterAttribute	Overloaded. Initializes a new instance of ContentImporterAttribute .



Public Properties

Name	Description
 CachedImportedData	Gets and sets the caching of the content during importation.
 DefaultProcessor	Gets or sets the name of the default processor for content read by this importer.
 DisplayName	Gets or sets the string representing the importer in a user interface.
 FileExtensions	Gets the supported file name extensions of the importer.
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentImporterAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterAttribute Constructor

Initializes a new instance of [ContentImporterAttribute](#).

Overload List

Name	Description
ContentImporterAttribute (String)	Initializes a new instance of ContentImporterAttribute and sets the file name extension supported by the importer.
ContentImporterAttribute (String[])	Initializes a new instance of ContentImporterAttribute and sets the file name extensions supported by the importer.

See Also

Reference

[ContentImporterAttribute Class](#)

[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterAttribute Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentImporterAttribute](#) and sets the file name extension supported by the importer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentImporterAttribute (  
    string fileExtension  
)
```

Parameters

fileExtension

The list of file name extensions supported by the importer. Prefix each extension with a '.'.

See Also

Reference

[ContentImporterAttribute Class](#)

[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterAttribute Constructor (String[])

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentImporterAttribute](#) and sets the file name extensions supported by the importer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentImporterAttribute (  
    string[] fileExtensions  
)
```

Parameters

fileExtensions

[[ParamArrayAttribute](#)] The list of file name extensions supported by the importer. Prefix each extension with a '.'.

See Also

Reference

[ContentImporterAttribute Class](#)










[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



Platforms Windows XP SP2, Windows Vista

ContentImporterAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ContentImporterAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterAttribute Properties

Public Properties

	Name	Description
	CacheImportedData	Gets and sets the caching of the content during importation.
	DefaultProcessor	Gets or sets the name of the default processor for content read by this importer.
	DisplayName	Gets or sets the string representing the importer in a user interface.
	FileExtensions	Gets the supported file name extensions of the importer.
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentImporterAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterAttribute.CacheImportedData Property

Note

This property is available only when developing for Windows.

Gets and sets the caching of the content during importation.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool CacheImportedData { get; set; }
```

Property Value

If **true**, imported content is cached in an intermediate file (managed by XNA). By default, caching is not enabled.

Caching provides significantly faster iteration times when changing the processing code for a file format. A good example would be a third-party digital content creation (DCC) tool that needs to execute before importation can occur. However, caching is not appropriate for all importers—for example, caching bulky content that is quick to import, such as bitmap images.

See Also

Reference

[ContentImporterAttribute Class](#)

[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterAttribute.DefaultProcessor Property

Note

This property is available only when developing for Windows.

Gets or sets the name of the default processor for content read by this importer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string DefaultProcessor { get; set; }
```

Property Value

Name of the processor to be used as the default.

See Also

Reference

[ContentImporterAttribute Class](#)

[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterAttribute.DisplayName Property

Note

This property is available only when developing for Windows.

Gets or sets the string representing the importer in a user interface. This name is not used by the content pipeline and should not be passed to the BuildAssets task (a custom [MSBuild](#) task used by XNA Game Studio). It is used for display purposes only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual string DisplayName { get; set; }
```

Property Value

Name displayed in the user interface.

See Also

Reference

[ContentImporterAttribute Class](#)

[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterAttribute.FileExtensions Property

Note

This property is available only when developing for Windows.

Gets the supported file name extensions of the importer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerable<string> FileExtensions { get; }
```

Property Value

An enumerated list of supported file name extensions. Extensions are prefixed by '.'.

See Also

Reference

[ContentImporterAttribute Class](#)

[ContentImporterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterContext Class

Note

This class is available only when developing for Windows.

Provides properties that define logging behavior for the importer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ContentImporterContext
```

See Also

Reference

[ContentImporterContext Members](#)




[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista







ContentImporterContext Members

The following tables list the members exposed by the ContentImporterContext type.



Public Properties

	Name	Description
	IntermediateDirectory	The absolute path to the root of the build intermediate (object) directory.
	Logger	Gets the logger for an importer.
	OutputDirectory	The absolute path to the root of the build output (binaries) directory.

Public Methods

	Name	Description
	AddDependency	Adds a dependency to the specified file.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[ContentImporterContext Class](#)



[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterContext Methods

Public Methods

	Name	Description
	AddDependency	Adds a dependency to the specified file.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentImporterContext Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterContext.AddDependency Method

Note

This method is available only when developing for Windows.

Adds a dependency to the specified file. This causes a rebuild of the file, when modified, on subsequent incremental builds.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void AddDependency (  
    string filename  
)
```

Parameters

filename

Name of an asset file.

RemarksDo not call this method for the main file passed to the [Import](#) method. It is only required for additional files needing to be read by the importer.

See Also

Reference

[ContentImporterContext Class](#)




[ContentImporterContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterContext Properties

Public Properties

	Name	Description
	IntermediateDirectory	The absolute path to the root of the build intermediate (object) directory.
	Logger	Gets the logger for an importer.
	OutputDirectory	The absolute path to the root of the build output (binaries) directory.

See Also

Reference

[ContentImporterContext Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentImporterContext.IntermediateDirectory Property

Note

This property is available only when developing for Windows.

The absolute path to the root of the build intermediate (object) directory.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string IntermediateDirectory { get; }
```

Property Value

The directory for storing temporary build files.

See Also

Reference

[ContentImporterContext Class](#)

[ContentImporterContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterContext.Logger Property

Note

This property is available only when developing for Windows.

Gets the logger for an importer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentBuildLogger Logger { get; }
```

Property Value

Logger that contains information on warnings and debug messages generated during importation.

See Also

Reference

[ContentImporterContext Class](#)

[ContentImporterContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentImporterContext.OutputDirectory Property

Note

This property is available only when developing for Windows.

The absolute path to the root of the build output (binaries) directory.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string OutputDirectory { get; }
```

Property Value

The output directory for the final build results.

See Also

Reference

[ContentImporterContext Class](#)

[ContentImporterContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentItem Class

Note

This class is available only when developing for Windows.

Provides properties that define various aspects of content stored using the intermediate file format of the XNA Framework.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class ContentItem
```

See Also

Reference

[ContentItem Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista




ContentItem Members

The following tables list the members exposed by the ContentItem type.






Public Constructors

	Name	Description
	ContentItem	Initializes a new instance of ContentItem .



Public Properties

	Name	Description
	Identity	Gets or sets the identity of the content item.
	Name	Gets or sets the name of the content item.
	OpaqueData	Gets the opaque data of the content item.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentItem Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentItem Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ContentItem](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentItem ()
```

See Also

Reference

[ContentItem Class](#)






[ContentItem Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentItem Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also




Reference

[ContentItem Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentItem Properties

Public Properties

	Name	Description
	Identity	Gets or sets the identity of the content item.
	Name	Gets or sets the name of the content item.
	OpaqueData	Gets the opaque data of the content item.

See Also

Reference

[ContentItem Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentItem.Identity Property

Note

This property is available only when developing for Windows.

Gets or sets the identity of the content item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentIdentity Identity { get; set; }
```

Property Value

The identity of the content item.

See Also

Reference

[ContentItem Class](#)

[ContentItem Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentItem.Name Property

Note

This property is available only when developing for Windows.

Gets or sets the name of the content item.

This name is case-sensitive.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

The name of the content item.

See Also

Reference

[ContentItem Class](#)

[ContentItem Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentItem.OpaqueData Property

Note

This property is available only when developing for Windows.

Gets the opaque data of the content item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public OpaqueDataDictionary OpaqueData { get; }
```

Property Value

The opaque data of the content item, stored as a set of key/value pairs.

See Also

Reference

[ContentItem Class](#)

[ContentItem Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessor Generic Class

Note

This generic class is available only when developing for Windows.

Provides a base class to use when developing custom processor components. All processors must derive from this class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentProcessor<TInput,TOutput> : IContentProcessor
```

See Also

Reference

[ContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista







ContentProcessor Members

The following tables list the members exposed by the ContentProcessor type.



Protected Constructors

	Name	Description
	ContentProcessor	Initializes a new instance of the ContentProcessor class.




Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Processes the specified input data and returns the result.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.InputType	Gets the expected object type of the input parameter to IContentProcessor.Process .
	Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.OutputType	Gets the object type returned by IContentProcessor.Process .
	Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.Process	Processes the specified input data and returns the result.

See Also

Reference

[ContentProcessor Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the **ContentProcessor** class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentProcessor ()
```

See Also

Reference

[ContentProcessor Generic Class](#)







[ContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentProcessor Methods




Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Processes the specified input data and returns the result.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.InputType	Gets the expected object type of the input parameter to IContentProcessor.Process .
 Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.OutputType	Gets the object type returned by IContentProcessor.Process .
 Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.Process	Processes the specified input data and returns the result.

See Also

Reference

[ContentProcessor Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.Process Method

Note

This method is available only when developing for Windows.

Processes the specified input data and returns the result.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private Object Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.Process (  
    Object input,  
    ContentProcessorContext context  
)
```

Parameters

input

Existing content object being processed.

context

Contains any required custom process parameters.

Return Value

The processed input.

See Also

Reference

[ContentProcessor Generic Class](#)

[ContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentProcessor.Process Method

Note

This method is available only when developing for Windows.

Processes the specified input data and returns the result.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract TOutput Process (  
    TInput input,  
    ContentProcessorContext context  
)
```

Parameters

input

Existing content object being processed.

context

Contains any required custom process parameters.

Return Value

A typed object representing the processed input.

See Also

Reference

[ContentProcessor Generic Class](#)

[ContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentProcessor Properties

See Also

Reference

[ContentProcessor Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessor.Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.InputType Property

Note

This property is available only when developing for Windows.

Gets the expected object type of the input parameter to [IContentProcessor.Process](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private Type Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.InputType { get; }
```

Property Value

Object type of the input parameter.

See Also

Reference

[ContentProcessor Generic Class](#)

[ContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessor.Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.OutputType Property

Note

This property is available only when developing for Windows.

Gets the object type returned by [IContentProcessor.Process](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private Type Microsoft.Xna.Framework.Content.Pipeline.IContentProcessor.OutputType { get; }
```

Property Value

Type of object returned by the processor.

See Also

Reference

[ContentProcessor Generic Class](#)

[ContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentProcessorAttribute Class

Note

This class is available only when developing for Windows.

Gets any existing content processor components.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class ContentProcessorAttribute : Attribute
```

Remarks

If you are developing a custom processor, it must use this attribute. The processor must also extend the [ContentProcessor](#) class.

See Also

Reference

[ContentProcessorAttribute Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista



ContentProcessorAttribute Members

The following tables list the members exposed by the ContentProcessorAttribute type.










Public Constructors

Name	Description
 ContentProcessorAttribute	Initializes an instance of ContentProcessorAttribute.



Public Properties

Name	Description
 DisplayName	Gets or sets the string representing the processor in a user interface.
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentProcessorAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorAttribute Constructor

Note

This constructor is available only when developing for Windows.

Initializes an instance of **ContentProcessorAttribute**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentProcessorAttribute ()
```

See Also

Reference

[ContentProcessorAttribute Class](#)










[ContentProcessorAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentProcessorAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[ContentProcessorAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorAttribute Properties

Public Properties

	Name	Description
	DisplayName	Gets or sets the string representing the processor in a user interface.
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentProcessorAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorAttribute.DisplayName Property

Note

This property is available only when developing for Windows.

Gets or sets the string representing the processor in a user interface. This name is not used by the content pipeline and should not be passed to the BuildAssets task (a custom [MSBuild](#) task used by XNA Game Studio). It is used for display purposes only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual string DisplayName { get; set; }
```

Property Value

Name displayed in the user interface.

See Also

Reference

[ContentProcessorAttribute Class](#)

[ContentProcessorAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext Class

Note

This class is available only when developing for Windows.

Provides access to custom processor parameters, methods for converting member data, and triggering nested builds.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ContentProcessorContext
```

Remarks

The context object, passed into a content processor, indicates any custom parameters or requests. For instance, a mesh processor could use this to force any textures used by the processor to be built when the mesh object is built.

See Also

Reference

[ContentProcessorContext Members](#)








[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista











ContentProcessorContext Members

The following tables list the members exposed by the ContentProcessorContext type.



Public Properties

Name	Description
 BuildConfiguration	Gets the name of the current content build configuration.
 IntermediateDirectory	Gets the path of the directory that will contain any intermediate files generated by the content processor.
 Logger	Gets the logger interface used for status messages or warnings.
 OutputDirectory	Gets the output path of the content processor.
 OutputFilename	Gets the output file name of the content processor.
 Parameters	Gets the collection of parameters used by the content processor.
 TargetPlatform	Gets the current content build target platform.

Public Methods

Name	Description
 AddDependency	Adds a dependency to the specified file.
 AddOutputFile	Add a file name to the list of related output files maintained by the build item.
 BuildAndLoadAsset	Overloaded. Initiates a nested build of the specified asset and then loads the result into memory.
 BuildAsset	Overloaded. Initiates a nested build of an asset.
 Convert	Overloaded. Converts a content item object using the specified content processor.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also











Reference

[ContentProcessorContext Class](#)



[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorContext Methods

Public Methods

	Name	Description
	AddDependency	Adds a dependency to the specified file.
	AddOutputFile	Add a file name to the list of related output files maintained by the build item.
	BuildAndLoadAsset	Overloaded. Initiates a nested build of the specified asset and then loads the result into memory.
	BuildAsset	Overloaded. Initiates a nested build of an asset.
	Convert	Overloaded. Converts a content item object using the specified content processor.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentProcessorContext Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorContext.AddDependency Method

Note

This method is available only when developing for Windows.

Adds a dependency to the specified file. This causes a rebuild of the file, when modified, on subsequent incremental builds.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void AddDependency (  
    string filename  
)
```

Parameters

filename

Name of an asset file.

Remarks Do not call this method for data that has already been loaded by an importer or passed to the [BuildAndLoadAsset](#) method. It is only required for additional files needing to be read by the processor.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentProcessorContext.AddOutputFile Method

Note

This method is available only when developing for Windows.

Add a file name to the list of related output files maintained by the build item. This allows tracking build items that build multiple output files.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void AddOutputFile (  
    string filename  
)
```

Parameters

filename

The name of the file.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.BuildAndLoadAsset Method

Initiates a nested build of the specified asset and then loads the result into memory.

Overload List

Name	Description
ContentProcessorContext.BuildAndLoadAsset (Generic ExternalReference, String)	Initiates a nested build of the specified asset and then loads the result into memory.
ContentProcessorContext.BuildAndLoadAsset (Generic ExternalReference, String, OpaqueDataDictionary, String)	Initiates a nested build of the specified asset and then loads the result into memory.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorContext.BuildAndLoadAsset Generic Method (Generic ExternalReference, String)

Note

This generic method is available only when developing for Windows.

Initiates a nested build of the specified asset and then loads the result into memory.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TOutput BuildAndLoadAsset<TInput,TOutput> (  
    ExternalReference<TInput> sourceAsset,  
    string processorName  
)
```

Type Parameters

TInput

Type of the input.

TOutput

Type of the converted output.

Parameters

sourceAsset

Reference to the source asset.

processorName

Optional processor for this content.

Return Value

Copy of the final converted content.

Remarks

An example of usage would be a mesh processor calling **BuildAndLoadAsset** to build any associated textures and replace the original .tga file references with an embedded copy of the converted texture.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentProcessorContext.BuildAndLoadAsset Generic Method (Generic ExternalReference, String, OpaqueDataDictionary, String)

Note

This generic method is available only when developing for Windows.

Initiates a nested build of the specified asset and then loads the result into memory.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TOutput BuildAndLoadAsset<TInput,TOutput> (  
    ExternalReference<TInput> sourceAsset,  
    string processorName,  
    OpaqueDataDictionary processorParameters,  
    string importerName  
)
```

Type Parameters

TInput

Type of the input.

TOutput

Type of the converted output.

Parameters

sourceAsset

Reference to the source asset.

processorName

Optional processor for this content.

processorParameters

Optional collection of named values available as input to the content processor.

importerName

Optional importer for this content.

Return Value

Copy of the final converted content.

Remarks

An example of usage would be a mesh processor calling **BuildAndLoadAsset** to build any associated textures and replace the original .tga file references with an embedded copy of the converted texture.

See Also

Conceptual

[Parameterized Processors](#)

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.BuildAsset Method

Initiates a nested build of an asset.

Overload List

Name	Description
ContentProcessorContext.BuildAsset (Generic ExternalReference, String)	Initiates a nested build of an additional asset.
ContentProcessorContext.BuildAsset (Generic ExternalReference, String, OpaqueDataDictionary, String, String)	Initiates a nested build of an additional asset.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorContext.BuildAsset Generic Method (Generic ExternalReference, String)

Note

This generic method is available only when developing for Windows.

Initiates a nested build of an additional asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference<TOutput> BuildAsset<TInput,TOutput> (  
    ExternalReference<TInput> sourceAsset,  
    string processorName  
)
```

Type Parameters

TInput

Type of the input.

TOutput

Type of the output.

Parameters

sourceAsset

Reference to the source asset.

processorName

Optional processor for this content.

Return Value

Reference to the final compiled content. The build work is not required to complete before returning. Therefore, this file may not be up to date when **BuildAsset** returns but it will be available for loading by the game at runtime.

Remarks

An example of usage for **BuildAsset** is being called by a mesh processor to request that any related textures used are also built, replacing the original TGA file references with new references to the converted texture files.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.BuildAsset Generic Method (Generic ExternalReference, String, OpaqueDataDictionary, String, String)

Note

This generic method is available only when developing for Windows.

Initiates a nested build of an additional asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference<TOutput> BuildAsset<TInput,TOutput> (
    ExternalReference<TInput> sourceAsset,
    string processorName,
    OpaqueDataDictionary processorParameters,
    string importerName,
    string assetName
)
```

Type Parameters

TInput

Type of the input.

TOutput

Type of the output.

Parameters

sourceAsset

Reference to the source asset.

processorName

Optional processor for this content.

processorParameters

Optional collection of named values available as input to the content processor.

importerName

Optional importer for this content.

assetName

Optional name of the final compiled content.

Return Value

Reference to the final compiled content. The build work is not required to complete before returning. Therefore, this file may not be up to date when **BuildAsset** returns but it will be available for loading by the game at runtime.

Remarks

An example of usage for **BuildAsset** is being called by a mesh processor to request that any related textures used are also built, replacing the original TGA file references with new references to the converted texture files.

See Also

Conceptual

[Parameterized Processors](#)

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.Convert Method

Converts a content item object using the specified content processor.

Overload List

Name	Description
ContentProcessorContext.Convert (TInput, String)	Converts a content item object using the specified content processor.
ContentProcessorContext.Convert (TInput, String, OpaqueDataDictionary)	Converts a content item object using the specified content processor.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorContext.Convert Generic Method (TInput, String)

Note

This generic method is available only when developing for Windows.

Converts a content item object using the specified content processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TOutput Convert<TInput,TOutput> (  
    TInput input,  
    string processorName  
)
```

Type Parameters

TInput

Type of the input content.

TOutput

Type of the converted output.

Parameters

input

Source content to be converted.

processorName

Optional processor for this content.

Return Value

Reference of the final converted content.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.Convert Generic Method (TInput, String, OpaqueDataDictionary)

Note

This generic method is available only when developing for Windows.

Converts a content item object using the specified content processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TOutput Convert<TInput,TOutput> (  
    TInput input,  
    string processorName,  
    OpaqueDataDictionary processorParameters  
)
```

Type Parameters

TInput

Type of the input content.

TOutput

Type of the converted output.

Parameters

input

Source content to be converted.

processorName

Optional processor for this content.

processorParameters

Optional parameters for the processor.

Return Value

Reference of the final converted content.

See Also

Reference

[ContentProcessorContext Class](#)








[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentProcessorContext Properties

Public Properties

	Name	Description
	BuildConfiguration	Gets the name of the current content build configuration.
	IntermediateDirectory	Gets the path of the directory that will contain any intermediate files generated by the content processor.
	Logger	Gets the logger interface used for status messages or warnings.
	OutputDirectory	Gets the output path of the content processor.
	OutputFilename	Gets the output file name of the content processor.
	Parameters	Gets the collection of parameters used by the content processor.
	TargetPlatform	Gets the current content build target platform.

See Also

Reference

[ContentProcessorContext Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ContentProcessorContext.BuildConfiguration Property

Note

This property is available only when developing for Windows.

Gets the name of the current content build configuration.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string BuildConfiguration { get; }
```

Property Value

Name of the build configuration.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.IntermediateDirectory Property

Note

This property is available only when developing for Windows.

Gets the path of the directory that will contain any intermediate files generated by the content processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string IntermediateDirectory { get; }
```

Property Value

The path of the directory that will contain any intermediate files that the content processor will produce.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.Logger Property

Note

This property is available only when developing for Windows.

Gets the logger interface used for status messages or warnings.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentBuildLogger Logger { get; }
```

Property Value

Logger interface used by the processor.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.OutputDirectory Property

Note

This property is available only when developing for Windows.

Gets the output path of the content processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string OutputDirectory { get; }
```

Property Value

The directory path of the output file that the content processor will produce.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.OutputFilename Property

Note

This property is available only when developing for Windows.

Gets the output file name of the content processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string OutputFilename { get; }
```

Property Value

The name of the output file that the content processor will produce.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.Parameters Property

Note

This property is available only when developing for Windows.

Gets the collection of parameters used by the content processor.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public OpaqueDataDictionary Parameters { get; }
```

Property Value

Collection of content processor parameters.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentProcessorContext.TargetPlatform Property

Note

This property is available only when developing for Windows.

Gets the current content build target platform.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TargetPlatform TargetPlatform { get; }
```

Property Value

Name of the target platform for the current content build.

See Also

Reference

[ContentProcessorContext Class](#)

[ContentProcessorContext Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ExternalReference Generic Class

Note

This generic class is available only when developing for Windows.

Specifies external references to a data file for the content item.

While the object model is instantiated, reference file names are absolute. When the file containing the external reference is serialized to disk, file names are relative to the file. This allows movement of the content tree to a different location without breaking internal links.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ExternalReference<T> : ContentItem
```

See Also

Reference

[ExternalReference Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista





ExternalReference Members

The following tables list the members exposed by the ExternalReference type.






Public Constructors

	Name	Description
	ExternalReference	Overloaded. Initializes a new instance of ExternalReference .



Public Properties

	Name	Description
	Filename	Gets and sets the file name of an ExternalReference .
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ExternalReference Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ExternalReference Constructor

Initializes a new instance of [ExternalReference](#).

Overload List

Name	Description
ExternalReference ()	Initializes a new instance of ExternalReference .
ExternalReference (String)	Initializes a new instance of ExternalReference .
ExternalReference (String, ContentId entity)	Initializes a new instance of ExternalReference , specifying the file path relative to another content item.

See Also

Reference

[ExternalReference Generic Class](#)

[ExternalReference Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ExternalReference Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ExternalReference](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference ()
```

See Also

Reference

[ExternalReference Generic Class](#)

[ExternalReference Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ExternalReference Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ExternalReference](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference (  
    string filename  
)
```

Parameters

filename

The name of the referenced file.

See Also

Reference

[ExternalReference Generic Class](#)

[ExternalReference Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ExternalReference Constructor (String, ContentIdentity)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [ExternalReference](#), specifying the file path relative to another content item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference (
    string filename,
    ContentIdentity relativeToContent
)
```

Parameters

filename

The name of the referenced file.

relativeToContent

The content that the path specified in *filename* is relative to.

Exceptions

Exception type	Condition
ArgumentNullException	<i>filename</i> , <i>relativeToContent</i> , or the SourceFilename property of <i>relativeToContent</i> is null .

Remarks

This method constructs a reference to the specified file. If the file name contains a relative path, it is made absolute based on the [SourceFilename](#) property of *relativeToContent*. This can be useful in a custom content processor for constructing references to texture or effect files relative to the [Identity](#) property of the mesh being processed.

See Also

Reference

[ExternalReference Generic Class](#)






[ExternalReference Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



Platforms Windows XP SP2, Windows Vista

ExternalReference Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[ExternalReference Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ExternalReference Properties

Public Properties

	Name	Description
	Filename	Gets and sets the file name of an ExternalReference .
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[ExternalReference Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ExternalReference.FileName Property

Note

This property is available only when developing for Windows.

Gets and sets the file name of an [ExternalReference](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string FileName { get; set; }
```

Property Value

The absolute file name of the external data reference.

See Also

Reference

[ExternalReference Generic Class](#)

[ExternalReference Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescriptionImporter Class

Note

This class is available only when developing for Windows.

Provides methods for reading .spritefont files for use in the Content Pipeline.

Note

As with most types of software, font files are licensed rather than sold. Font licenses vary from vendor to vendor, but most don't allow redistribution of the fonts, and that includes redistribution of reproductions such as bitmaps containing the rasterized character set. This is even true of many of the licenses covering fonts that Microsoft supplies with applications and Windows. Be careful, therefore, to ensure that you have the required license rights to redistribute any font you include as a bitmap containing the rasterized character set in your game!

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentImporterAttribute(".spritefont")]  
public class FontDescriptionImporter : ContentImporter<FontDescription>
```

See Also

Reference

[FontDescriptionImporter Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista







FontDescriptionImporter Members

The following tables list the members exposed by the FontDescriptionImporter type.



Public Constructors

Name	Description
 FontDescriptionImporter	Initializes a new instance of FontDescriptionImporter.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Import	Called by the XNA Framework when importing a .spritefont file to be used as a game asset. This is the method called by the XNA Framework when an asset is to be imported into an object that can be recognized by the Content Pipeline.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontDescriptionImporter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

FontDescriptionImporter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **FontDescriptionImporter**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontDescriptionImporter ()
```

See Also

Reference

[FontDescriptionImporter Class](#)







[FontDescriptionImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

FontDescriptionImporter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Import	Called by the XNA Framework when importing a .spritefont file to be used as a game asset. This is the method called by the XNA Framework when an asset is to be imported into an object that can be recognized by the Content Pipeline.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontDescriptionImporter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

FontDescriptionImporter.Import Method

Note

This method is available only when developing for Windows.

Called by the XNA Framework when importing a .spritefont file to be used as a game asset. This is the method called by the XNA Framework when an asset is to be imported into an object that can be recognized by the Content Pipeline.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override FontDescription Import (
    string filename,
    ContentImporterContext context
)
```

Parameters

filename

Name of a game asset file.

context

Contains information for importing a game asset, such as a logger interface.

Return Value

Resulting game asset.

Exceptions

Exception type	Condition
ArgumentNullException	<i>filename</i> is null .
SecurityException	Sufficient permissions are not available to access <i>filename</i> .
FileNotFoundException	The file identified by the <i>filename</i> does not exist.
UriFormatException	The format of <i>filename</i> is not correct.
InvalidContentException	The XML in <i>filename</i> is either badly formed or does not represent a FontDescription object.

See Also

Reference

[FontDescriptionImporter Class](#)

[FontDescriptionImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

IContentImporter Interface

Note

This interface is available only when developing for Windows.

Accesses a statically typed [ContentImporter](#) instance from generic code using dynamic typing.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public interface IContentImporter
```

See Also

Reference

[IContentImporter Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

IContentImporter Members

The following tables list the members exposed by the IContentImporter type.

Public Methods

	Name	Description
	Import	Imports an asset from the specified file.

See Also


Reference

[IContentImporter Interface](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

IContentImporter Methods

Public Methods

	Name	Description
	Import	Imports an asset from the specified file.

See Also

Reference

[IContentImporter Interface](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

IContentImporter.Import Method

Note

This method is available only when developing for Windows.

Imports an asset from the specified file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object Import (  
    string filename,  
    ContentImporterContext context  
)
```

Parameters

filename

Name of the game asset file.

context

A [ContentImporterContext](#) class containing information for importing a game asset, such as a logger interface.

Return Value

Resulting game asset.

See Also

Reference

[IContentImporter Interface](#)

[IContentImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

IContentProcessor Interface

Note

This interface is available only when developing for Windows.

Provides methods and properties for accessing a statically typed [ContentProcessor](#) subclass, using dynamically typed object data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public interface IContentProcessor
```

See Also

Reference

[IContentProcessor Members](#)



[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista


IContentProcessor Members

The following tables list the members exposed by the IContentProcessor type.

Public Properties

	Name	Description
	InputType	Gets the expected object type of the input parameter to IContentProcessor.Process .
	OutputType	Gets the object type returned by IContentProcessor.Process .

Public Methods

	Name	Description
	Process	Processes the specified input data and returns the result.

See Also


Reference

[IContentProcessor Interface](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

IContentProcessor Methods

Public Methods

	Name	Description
	Process	Processes the specified input data and returns the result.

See Also

Reference

[IContentProcessor Interface](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

IContentProcessor.Process Method

Note

This method is available only when developing for Windows.

Processes the specified input data and returns the result.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object Process (  
    Object input,  
    ContentProcessorContext context  
)
```

Parameters

input

Existing content object being processed.

context

Contains any required custom process parameters.

Return Value

An object representing the processed input.

See Also

Reference

[IContentProcessor Interface](#)



[IContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

IContentProcessor Properties

Public Properties

	Name	Description
	InputType	Gets the expected object type of the input parameter to IContentProcessor.Process .
	OutputType	Gets the object type returned by IContentProcessor.Process .

See Also

Reference

[IContentProcessor Interface](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

IContentProcessor.InputType Property

Note

This property is available only when developing for Windows.

Gets the expected object type of the input parameter to [IContentProcessor.Process](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract Type InputType { get; }
```

Property Value

Object type of the input parameter.

See Also

Reference

[IContentProcessor Interface](#)

[IContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IContentProcessor.OutputType Property

Note

This property is available only when developing for Windows.

Gets the object type returned by [IContentProcessor.Process](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract Type OutputType { get; }
```

Property Value

Type of object returned by the processor.

See Also

Reference

[IContentProcessor Interface](#)

[IContentProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

InvalidContentException Class

Note

This class is available only when developing for Windows.

Thrown when errors are encountered in content during processing.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidContentException : Exception
```

Remarks

InvalidContentException is an exception that any importer or content processor can throw to indicate a problem with the current content file. Exceptions of this type are presented differently from all other exceptions the pipeline throws. Exceptions of this type indicate a problem in the content. Other exceptions the content pipeline throws indicate a problem with the pipeline code itself. The **InvalidContentException** class has properties that identify the content file with the problem, if available.

See Also

Reference

[InvalidContentException Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista









InvalidContentException Members

The following tables list the members exposed by the InvalidContentException type.


Public Constructors

Name	Description
 InvalidContentException	Overloaded. Creates an instance of InvalidContentException .







Public Properties

Name	Description
 ContentIdentity	Gets or sets the identity of the content item that caused the exception.
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	When overridden in a derived class, returns information about the exception.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[InvalidContentException Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

InvalidContentException Constructor

Creates an instance of [InvalidContentException](#).

Overload List

Name	Description
InvalidContentException ()	Initializes a new instance of the InvalidContentException class
InvalidContentException (SerializationInfo, StreamingContext)	Initializes a new instance of the InvalidContentException class with information on serialization and streaming context for the related content item.
InvalidContentException (String)	Initializes a new instance of the InvalidContentException class with the specified error message.
InvalidContentException (String, ContentIdentity)	Initializes a new instance of the InvalidContentException class with the specified error message and the identity of the content throwing the exception.
InvalidContentException (String, ContentIdentity, Exception)	Initializes a new instance of the InvalidContentException class with the specified error message, the identity of the content throwing the exception, and a reference to the inner exception that is the cause of this exception.
InvalidContentException (String, Exception)	Initializes a new instance of the InvalidContentException class with the specified error message and a reference to the inner exception that is the cause of this exception.

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

InvalidContentException Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [InvalidContentException](#) class

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public InvalidContentException ()
```

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

InvalidContentException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [InvalidContentException](#) class with information on serialization and streaming context for the related content item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected InvalidContentException (  
    SerializationInfo serializationInfo,  
    StreamingContext streamingContext  
)
```

Parameters

serializationInfo

Information necessary for serialization and deserialization of the content item.

streamingContext

Information necessary for the source and destination of a given serialized stream. Also provides an additional caller-defined context.

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

InvalidContentException Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [InvalidContentException](#) class with the specified error message.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public InvalidContentException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

InvalidContentException Constructor (String, ContentIdentity)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [InvalidContentException](#) class with the specified error message and the identity of the content throwing the exception.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public InvalidContentException (  
    string message,  
    ContentIdentity contentIdentity  
)
```

Parameters

message

A message that describes the error.

contentIdentity

Information about the content item that caused this error, including the file name. In some cases, a location within the file (of the problem) is specified.

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

InvalidContentException Constructor (String, ContentIdentity, Exception)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [InvalidContentException](#) class with the specified error message, the identity of the content throwing the exception, and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public InvalidContentException (
    string message,
    ContentIdentity contentIdentity,
    Exception innerException
)
```

Parameters

message

A message that describes the error.

contentIdentity

Information about the content item that caused this error, including the file name. In some cases, a location within the file (of the problem) is specified.

innerException

The exception that is the cause of the current exception. If *innerException* is not a null reference, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

InvalidContentException Constructor (String, Exception)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [InvalidContentException](#) class with the specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public InvalidContentException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The exception that is the cause of the current exception. If *innerException* is not a null reference, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[InvalidContentException Class](#)







[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

InvalidContentException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	When overridden in a derived class, returns information about the exception.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[InvalidContentException Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

InvalidContentException.GetObjectData Method

Note

This method is available only when developing for Windows.

When overridden in a derived class, returns information about the exception.

In addition to the base behavior, this method provides serialization functionality.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override void GetObjectData (  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameters

info

Information necessary for serialization and deserialization of the content item.

context

Information necessary for the source and destination of a given serialized stream. Also provides an additional caller-defined context.

See Also

Reference

[InvalidContentException Class](#)









[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)


PlatformsWindows XP SP2, Windows Vista

InvalidContentException Properties

Public Properties

	Name	Description
	ContentIdentity	Gets or sets the identity of the content item that caused the exception.
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[InvalidContentException Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

InvalidContentException.ContentIdentity Property

Note

This property is available only when developing for Windows.

Gets or sets the identity of the content item that caused the exception.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentIdentity ContentIdentity { get; set; }
```

Property Value

The identity of the bad content item.

See Also

Reference

[InvalidContentException Class](#)

[InvalidContentException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary Generic Class

Note

This generic class is available only when developing for Windows.

Base class for dictionaries that map string identifiers to data values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class NamedValueDictionary<T> : IDictionary<string, T>, ICollection<KeyValuePair<string, T>>, IEnumerable<KeyValuePair<string, T>>, IEnumerable
```

See Also

Reference

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary Members

The following tables list the members exposed by the NamedValueDictionary type.

Public Constructors

Name	Description
NamedValueDictionary	Initializes an instance of NamedValueDictionary .

Public Properties

Name	Description
Count	Gets the number of items in the dictionary.
Item	Gets or sets the specified item.
Keys	Gets all keys contained in the dictionary.
Values	Gets all values contained in the dictionary.

Protected Properties

Name	Description
DefaultSerializerType	Specifies the type hint for the intermediate serializer.

Public Methods

Name	Description
Add	Adds the specified key and value to the dictionary.
Clear	Removes all keys and values from the dictionary.
ContainsKey	Determines whether the specified key is present in the dictionary.
Equals	(Inherited from Object .)
GetEnumerator	Gets an enumerator that iterates through items in a dictionary.
GetHashCode	(Inherited from Object .)
GetType	(Inherited from Object .)
ReferenceEquals	(Inherited from Object .)
Remove	Removes the specified key and value from the dictionary.
ToString	(Inherited from Object .)
TryGetValue	Gets the value associated with the specified key.

Protected Methods

Name	Description
AddItem	Adds an element to the dictionary.
ClearItems	Removes all elements from the dictionary.
Finalize	(Inherited from Object .)
MemberwiseClone	(Inherited from Object .)
RemoveItem	Removes the specified element from the dictionary.
SetItem	Modifies the value of an existing element.

Explicit Interface Implementations

Name	Description
System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.IsReadOnly	Gets a value indicating if this object is read-only.
System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Add	Adds an item to the collection.
System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Contains	Determines whether the collection contains a specific value.
System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.CopyTo	Copies the elements of the collection to an array, starting at a specified index.
System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that can iterate through this collection.
System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Remove	Removes the first occurrence of the specified object from the collection.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

NamedValueDictionary Constructor

Note

This constructor is available only when developing for Windows.

Initializes an instance of [NamedValueDictionary](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public NamedValueDictionary ()
```

See Also

Reference

[NamedValueDictionary Generic Class](#)












[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)







PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary Methods







Public Methods

Name	Description
 Add	Adds the specified key and value to the dictionary.
 Clear	Removes all keys and values from the dictionary.
 ContainsKey	Determines whether the specified key is present in the dictionary.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator that iterates through items in a dictionary.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the specified key and value from the dictionary.
 ToString	(Inherited from Object .)
 TryGetValue	Gets the value associated with the specified key.

Protected Methods

Name	Description
 AddItem	Adds an element to the dictionary.
 ClearItems	Removes all elements from the dictionary.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes the specified element from the dictionary.
 SetItem	Modifies the value of an existing element.

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.IsReadOnly	Gets a value indicating if this object is read-only.
 System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Add	Adds an item to the collection.
 System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Contains	Determines whether the collection contains a specific value.
 System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.CopyTo	Copies the elements of the collection to an array, starting at a specified index.
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that can iterate through this collection.
 System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Remove	Removes the first occurrence of the specified object from the collection.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

NamedValueDictionary.Add Method

Note

This method is available only when developing for Windows.

Adds the specified key and value to the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Add (  
    string key,  
    T value  
)
```

Parameters

key

Identity of the key of the new data pair.

value

The value of the new data pair.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

NamedValueDictionary.AddItem Method

Note

This method is available only when developing for Windows.

Adds an element to the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual void AddItem (  
    string key,  
    T value  
)
```

Parameters

key

Identity of the key of the new element.

value

The value of the new element.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

NamedValueDictionary.Clear Method

Note

This method is available only when developing for Windows.

Removes all keys and values from the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Clear ()
```

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.ClearItems Method

Note

This method is available only when developing for Windows.

Removes all elements from the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual void ClearItems ()
```

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.ContainsKey Method

Note

This method is available only when developing for Windows.

Determines whether the specified key is present in the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool ContainsKey (  
    string key  
)
```

Parameters

key

Identity of a key.

Return Value

true if the value is present; **false** otherwise.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.GetEnumerator Method

Note

This method is available only when developing for Windows.

Gets an enumerator that iterates through items in a dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerable<KeyValuePair<string, T>> GetEnumerator ()
```

Return Value

Enumerator for iterating through the dictionary.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.Remove Method

Note

This method is available only when developing for Windows.

Removes the specified key and value from the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Remove (  
    string key  
)
```

Parameters

key

Identity of the key to be removed.

Return Value

true if the value is present; **false** otherwise.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.RemoveItem Method

Note

This method is available only when developing for Windows.

Removes the specified element from the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual bool RemoveItem (  
    string key  
)
```

Parameters

key

Identity of the key of the data pair to be removed.

Return Value

true if the value is present; **false** otherwise.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.SetItem Method

Note

This method is available only when developing for Windows.

Modifies the value of an existing element.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual void SetItem (  
    string key,  
    T value  
)
```

Parameters

key

Identity of the element to be modified.

value

The value to be set.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Add Method

Note

This method is available only when developing for Windows.

Adds an item to the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Add (  
    KeyValuePair<string, T> item  
)
```

Parameters

item

The item to add to the collection.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Contains Method

Note

This method is available only when developing for Windows.

Determines whether the collection contains a specific value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Contains (
    KeyValuePair<string, T> item
)
```

Parameters

item

The object to locate in the collection.

Return Value

true if the collection contains the object; **false** otherwise.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.CopyTo Method

Note

This method is available only when developing for Windows.

Copies the elements of the collection to an array, starting at a specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.CopyTo (
    KeyValuePair<string, T>[] array,
    int arrayIndex
)
```

Parameters

array

The index at which to begin the copy.

arrayIndex

The destination array.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Remove Method

Note

This method is available only when developing for Windows.

Removes the first occurrence of the specified object from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.Remove (
    KeyValuePair<string, T> item
)
```

Parameters

item

The item to remove from the collection.

Return Value

true if the item was successfully removed from the collection; **false** otherwise.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.IEnumerable.GetEnumerator Method

Note

This method is available only when developing for Windows.

Returns an enumerator that can iterate through this collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

An enumerator that can iterate through this collection

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.TryGetValue Method

Note

This method is available only when developing for Windows.

Gets the value associated with the specified key.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool TryGetValue (  
    string key,  
    out T value  
)
```

Parameters

key

Identity of the key of the element whose value is to be retrieved.

value

[[OutAttribute](#)] The current value of the element.

Return Value

true if the value is present; **false** otherwise.

Remarks

If the key was not found, *value* gets the appropriate default value for the type **T**. For example, 0 (zero) for integer types, **false** for Boolean types, and a **null** reference for reference types.

See Also

Reference

[NamedValueDictionary Generic Class](#)





[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)


PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary Properties

Public Properties

	Name	Description
	Count	Gets the number of items in the dictionary.
	Item	Gets or sets the specified item.
	Keys	Gets all keys contained in the dictionary.
	Values	Gets all values contained in the dictionary.

Protected Properties

	Name	Description
	DefaultSerializerType	Specifies the type hint for the intermediate serializer.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

NamedValueDictionary.Count Property

Note

This property is available only when developing for Windows.

Gets the number of items in the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of items.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.DefaultSerializerType Property

Note

This property is available only when developing for Windows.

Specifies the type hint for the intermediate serializer. Values of this type do not store an explicit type attribute in the related XML source.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal virtual Type DefaultSerializerType { get; }
```

Property Value

The type used by the serializer.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.Item Property

Note

This property is available only when developing for Windows.

Gets or sets the specified item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T this [
    string key
] { get; set; }
```

Property Value

Identity of a key.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.Keys Property

Note

This property is available only when developing for Windows.

Gets all keys contained in the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ICollection<string> Keys { get; }
```

Property Value

Collection of keys for the dictionary.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair> Property

Note

This property is available only when developing for Windows.

Gets a value indicating if this object is read-only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<System.Collections.Generic.KeyValuePair<System.String,T>>.IsReadOnly { get; }
```

Property Value

true if the object is read-only; **false** otherwise.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NamedValueDictionary.Values Property

Note

This property is available only when developing for Windows.

Gets all values contained in the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ICollection<T> Values { get; }
```

Property Value

Collection of values for the dictionary.

See Also

Reference

[NamedValueDictionary Generic Class](#)

[NamedValueDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary Class

Note

This class is available only when developing for Windows.

Provides properties that define opaque data for a game asset.

Opaque data is any data that is not explicitly supported by the XNA object model. In addition, the opaque data can be of any managed data type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class OpaqueDataDictionary : NamedValueDictionary<Object>
```

See Also

Reference

[OpaqueDataDictionary Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista





OpaqueDataDictionary Members

The following tables list the members exposed by the OpaqueDataDictionary type.


Public Constructors

Name	Description
 OpaqueDataDictionary	Initializes a new instance of OpaqueDataDictionary .
















Public Properties

Name	Description
 Count	(Inherited from NamedValueDictionary .)
 Item	(Inherited from NamedValueDictionary .)
 Keys	(Inherited from NamedValueDictionary .)
 Values	(Inherited from NamedValueDictionary .)





Protected Properties

Name	Description
 DefaultSerializerType	Specifies the type hint for the intermediate serializer.

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 AddItem	Overloaded. Adds an element to the dictionary.
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetContentAsXml	Gets the data contents and returns it as a single XML string.
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetValue	Gets the value of the specified key/value pair of the asset.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 SetItem	Overloaded. Modifies the value of an existing element.
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 ClearItems	Removes all elements from the dictionary.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes the specified element from the dictionary.

See Also

Reference

[OpaqueDataDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

OpaqueDataDictionary Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [OpaqueDataDictionary](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public OpaqueDataDictionary ()
```

See Also

Reference

[OpaqueDataDictionary Class](#)
















[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)





PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary Methods

Public Methods

	Name	Description
	Add	(Inherited from NamedValueDictionary .)
	AddItem	Overloaded. Adds an element to the dictionary.
	Clear	(Inherited from NamedValueDictionary .)
	ContainsKey	(Inherited from NamedValueDictionary .)
	Equals	(Inherited from Object .)
	GetContentAsXml	Gets the data contents and returns it as a single XML string.
	GetEnumerator	(Inherited from NamedValueDictionary .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetValue	Gets the value of the specified key/value pair of the asset.
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from NamedValueDictionary .)
	SetItem	Overloaded. Modifies the value of an existing element.
	ToString	(Inherited from Object .)
	TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

	Name	Description
	ClearItems	Removes all elements from the dictionary.
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	Removes the specified element from the dictionary.

See Also

Reference

[OpaqueDataDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

OpaqueDataDictionary.AddItem Method

Adds an element to the dictionary.

Overload List

Name	Description
OpaqueDataDictionary.AddItem (String, Object)	Adds an element to the dictionary.
OpaqueDataDictionary.AddItem (String, Object)	(Inherited from NamedValueDictionary.)

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

OpaqueDataDictionary.AddItem Method (String, Object)

Note

This method is available only when developing for Windows.

Adds an element to the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void AddItem (  
    string key,  
    Object value  
)
```

Parameters

key

Identity of the key of the new element.

value

The value of the new element.

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary.ClearItems Method

Note

This method is available only when developing for Windows.

Removes all elements from the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void ClearItems ()
```

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary.GetContentAsXml Method

Note

This method is available only when developing for Windows.

Gets the data contents and returns it as a single XML string. This format is useful for doing deep equality comparison or object hashing.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string GetContentAsXml ()
```

Return Value

Data contents of the entry.

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary.GetValue Generic Method

Note

This generic method is available only when developing for Windows.

Gets the value of the specified key/value pair of the asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T GetValue<T> (  
    string key,  
    T defaultValue  
)
```

Type Parameters

T

Parameters

key

The name of the key.

defaultValue

The value to return if the key cannot be found. This can be **null** for reference types, 0 for primitive types, and a zero-filled structure for structure types.

Return Value

The value of the retrieved opaque data.

Remarks

GetValue looks only for values of a specific type. If the entry is of a different type, the *defaultvalue* is returned.

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary.RemoveItem Method

Note

This method is available only when developing for Windows.

Removes the specified element from the dictionary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override bool RemoveItem (  
    string key  
)
```

Parameters

key

Identity of the key of the data pair to be removed.

Return Value

true if the value is present; **false** otherwise.

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary.SetItem Method

Modifies the value of an existing element.

Overload List

Name	Description
OpaqueDataDictionary.SetItem (String, Object)	Modifies the value of an existing element.
OpaqueDataDictionary.SetItem (String, Object)	(Inherited from NamedValueDictionary .)

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

OpaqueDataDictionary.SetItem Method (String, Object)

Note

This method is available only when developing for Windows.

Modifies the value of an existing element.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void SetItem (  
    string key,  
    Object value  
)
```

Parameters

key

Identity of the element to be modified.

value

The value to be set.

See Also

Reference

[OpaqueDataDictionary Class](#)





[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)


PlatformsWindows XP SP2, Windows Vista

OpaqueDataDictionary Properties

Public Properties

	Name	Description
	Count	(Inherited from NamedValueDictionary .)
	Item	(Inherited from NamedValueDictionary .)
	Keys	(Inherited from NamedValueDictionary .)
	Values	(Inherited from NamedValueDictionary .)

Protected Properties

	Name	Description
	DefaultSerializerType	Specifies the type hint for the intermediate serializer.

See Also

Reference

[OpaqueDataDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

OpaqueDataDictionary.DefaultSerializerType Property

Note

This property is available only when developing for Windows.

Specifies the type hint for the intermediate serializer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal override Type DefaultSerializerType { get; }
```

Property Value

The type used by the serializer.

See Also

Reference

[OpaqueDataDictionary Class](#)

[OpaqueDataDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner Class

Note

This class is available only when developing for Windows.

Implements a scanner object containing the available importers and processors for an application. Designed for internal use only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class PipelineComponentScanner
```

See Also

Reference

[PipelineComponentScanner Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista










PipelineComponentScanner Members

The following tables list the members exposed by the PipelineComponentScanner type.







Public Constructors

Name	Description
 PipelineComponentScanner	Initializes a new instance of PipelineComponentScanner.



Public Properties

Name	Description
 Errors	Gets the list of error messages produced by the last call to Update .
 ImporterAttributes	Gets a dictionary that maps importer names to their associated metadata attributes.
 ImporterNames	Gets the names of all available importers.
 ImporterOutputTypes	Gets a dictionary that maps importer names to the fully qualified name of their return types.
 ProcessorAttributes	Gets a dictionary that maps processor names to their associated metadata attributes.
 ProcessorInputTypes	Gets a dictionary that maps processor names to the fully qualified name of supported input types.
 ProcessorNames	Gets the names of all available importers.
 ProcessorOutputTypes	Gets a dictionary that maps processor names to the fully qualified name of their output types.
 ProcessorParameters	A collection of supported processor parameters.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Update	Overloaded. Updates the scanner object with the latest available assembly states.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PipelineComponentScanner Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineComponentScanner Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **PipelineComponentScanner**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PipelineComponentScanner ()
```

See Also

Reference

[PipelineComponentScanner Class](#)







[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Update	Overloaded. Updates the scanner object with the latest available assembly states.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PipelineComponentScanner Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineComponentScanner.Update Method

Updates the scanner object with the latest available assembly states.

Overload List

Name	Description
PipelineComponentScanner.Update (Generic IEnumerable)	Updates the scanner object with the latest available assembly states.
PipelineComponentScanner.Update (Generic IEnumerable, Generic IEnumerable)	Updates the scanner object with the latest available assembly states.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineComponentScanner.Update Method (Generic IEnumerable)

Note

This method is available only when developing for Windows.

Updates the scanner object with the latest available assembly states.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Update (  
    IEnumerable<string> pipelineAssemblies  
)
```

Parameters

pipelineAssemblies

Enumerated list of available assemblies.

Return Value

true if an actual scan was required, indicating the collection contents may have changed. **false** if no assembly changes were detected since the previous call.

Remarks

After **Update** is called, the [Importer*](#) and [Processor*](#) properties contain any new information. If any errors occurred during the scan, they are stored in the [Errors](#) property.

Since **Update** uses assembly file time stamps, trivial reject situations (where nothing has changed) are detected. Therefore, **Update** can be called frequently.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista

PipelineComponentScanner.Update Method (Generic IEnumerable, Generic IEnumerable)

Note

This method is available only when developing for Windows.

Updates the scanner object with the latest available assembly states.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Update (  
    IEnumerable<string> pipelineAssemblies,  
    IEnumerable<string> pipelineAssemblyDependencies  
)
```

Parameters

pipelineAssemblies

Enumerated list of available assemblies.

pipelineAssemblyDependencies

Enumerated list of dependent assemblies.

Return Value

true if an actual scan was required, indicating the collection contents may have changed. **false** if no assembly changes were detected since the previous call.

See Also

Reference

[PipelineComponentScanner Class](#)










[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner Properties

Public Properties

	Name	Description
	Errors	Gets the list of error messages produced by the last call to Update .
	ImporterAttributes	Gets a dictionary that maps importer names to their associated metadata attributes.
	ImporterNames	Gets the names of all available importers.
	ImporterOutputTypes	Gets a dictionary that maps importer names to the fully qualified name of their return types.
	ProcessorAttributes	Gets a dictionary that maps processor names to their associated metadata attributes.
	ProcessorInputTypes	Gets a dictionary that maps processor names to the fully qualified name of supported input types.
	ProcessorNames	Gets the names of all available importers.
	ProcessorOutputTypes	Gets a dictionary that maps processor names to the fully qualified name of their output types.
	ProcessorParameters	A collection of supported processor parameters.

See Also

Reference

[PipelineComponentScanner Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineComponentScanner.Errors Property

Note

This property is available only when developing for Windows.

Gets the list of error messages produced by the last call to [Update](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IList<string> Errors { get; }
```

Property Value

Collection of the latest error descriptions.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ImporterAttributes Property

Note

This property is available only when developing for Windows.

Gets a dictionary that maps importer names to their associated metadata attributes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IDictionary<string, ContentImporterAttribute> ImporterAttributes { get; }
```

Property Value

Importer names and their related metadata values.

Remarks

Call **ImporterAttributes** to look up the [DisplayName](#) value for an importer object and which file extensions are supported.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ImporterNames Property

Note

This property is available only when developing for Windows.

Gets the names of all available importers.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerable<string> ImporterNames { get; }
```

Property Value

List of available importer names.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ImporterOutputTypes Property

Note

This property is available only when developing for Windows.

Gets a dictionary that maps importer names to the fully qualified name of their return types.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IDictionary<string, string> ImporterOutputTypes { get; }
```

Property Value

Importer names and their related return types.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ProcessorAttributes Property

Note

This property is available only when developing for Windows.

Gets a dictionary that maps processor names to their associated metadata attributes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IDictionary<string, ContentProcessorAttribute> ProcessorAttributes { get; }
```

Property Value

Processor names and their related metadata values.

Remarks

Call **ProcessorAttributes** to look up the [DisplayName](#) value for an processor object.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ProcessorInputTypes Property

Note

This property is available only when developing for Windows.

Gets a dictionary that maps processor names to the fully qualified name of supported input types.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IDictionary<string, string> ProcessorInputTypes { get; }
```

Property Value

Processor names and the input types they support.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ProcessorNames Property

Note

This property is available only when developing for Windows.

Gets the names of all available importers.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerable<string> ProcessorNames { get; }
```

Property Value

List of available processor names.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ProcessorOutputTypes Property

Note

This property is available only when developing for Windows.

Gets a dictionary that maps processor names to the fully qualified name of their output types.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IDictionary<string, string> ProcessorOutputTypes { get; }
```

Property Value

Processor names and their related output types.

See Also

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineComponentScanner.ProcessorParameters Property

Note

This property is available only when developing for Windows.

A collection of supported processor parameters.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IDictionary<string, ProcessorParameterCollection> ProcessorParameters { get; }
```

Property Value

Collection of processor parameters.

See Also

Conceptual

[Parameterized Processors](#)

Reference

[PipelineComponentScanner Class](#)

[PipelineComponentScanner Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineException Class

Note

This class is available only when developing for Windows.

Thrown when errors are encountered during a content pipeline build.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[SerializableAttribute]  
public class PipelineException : Exception
```

See Also

Reference

[PipelineException Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista








PipelineException Members

The following tables list the members exposed by the PipelineException type.


Public Constructors

Name	Description
 PipelineException	Overloaded. Creates an instance of PipelineException .







Public Properties

Name	Description
 Data	(Inherited from Exception .)
 HelpLink	(Inherited from Exception .)
 InnerException	(Inherited from Exception .)
 Message	(Inherited from Exception .)
 Source	(Inherited from Exception .)
 StackTrace	(Inherited from Exception .)
 TargetSite	(Inherited from Exception .)



Protected Properties

Name	Description
 HResult	(Inherited from Exception .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetBaseException	(Inherited from Exception .)
 GetHashCode	(Inherited from Object .)
 GetObjectData	(Inherited from Exception .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PipelineException Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineException Constructor

Creates an instance of [PipelineException](#).

Overload List

Name	Description
PipelineException ()	Creates an instance of PipelineException .
PipelineException (SerializationInfo, StreamingContext)	Creates an instance of PipelineException with information on serialization and streaming context for the related content item.
PipelineException (String)	Initializes a new instance of the PipelineException class with the specified error message.
PipelineException (String, Exception)	Initializes a new instance of the PipelineException class with the specified error message and a reference to the inner exception that is the cause of this exception.
PipelineException (String, Object[])	Initializes a new instance of the PipelineException class with the specified error message.

See Also

Reference

[PipelineException Class](#)

[PipelineException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineException Constructor ()

Note

This constructor is available only when developing for Windows.

Creates an instance of [PipelineException](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PipelineException ()
```

See Also

Reference

[PipelineException Class](#)

[PipelineException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineException Constructor (SerializationInfo, StreamingContext)

Note

This constructor is available only when developing for Windows.

Creates an instance of [PipelineException](#) with information on serialization and streaming context for the related content item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected PipelineException (  
    SerializationInfo serializationInfo,  
    StreamingContext streamingContext  
)
```

Parameters

serializationInfo

Information necessary for serialization and deserialization of the content item.

streamingContext

Information necessary for the source and destination of a given serialized stream. Also provides an additional caller-defined context.

See Also

Reference

[PipelineException Class](#)

[PipelineException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineException Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [PipelineException](#) class with the specified error message.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PipelineException (  
    string message  
)
```

Parameters

message

A message that describes the error.

See Also

Reference

[PipelineException Class](#)

[PipelineException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineException Constructor (String, Exception)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [PipelineException](#) class with the specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PipelineException (  
    string message,  
    Exception innerException  
)
```

Parameters

message

A message that describes the error.

innerException

The exception that is the cause of the current exception. If *innerException* is not a null reference, the current exception is raised in a catch block that handles the inner exception.

See Also

Reference

[PipelineException Class](#)

[PipelineException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PipelineException Constructor (String, Object[])

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [PipelineException](#) class with the specified error message.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PipelineException (  
    string message,  
    Object[] messageArgs  
)
```

Parameters

message

A message that describes the error.

messageArgs

[[ParamArrayAttribute](#)] Array of strings specifying message-related arguments.

See Also

Reference

[PipelineException Class](#)







[PipelineException Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

PipelineException Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetBaseException	(Inherited from Exception .)
	GetHashCode	(Inherited from Object .)
	GetObjectData	(Inherited from Exception .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[PipelineException Class](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PipelineException Properties

Public Properties

	Name	Description
	Data	(Inherited from Exception .)
	HelpLink	(Inherited from Exception .)
	InnerException	(Inherited from Exception .)
	Message	(Inherited from Exception .)
	Source	(Inherited from Exception .)
	StackTrace	(Inherited from Exception .)
	TargetSite	(Inherited from Exception .)

Protected Properties

	Name	Description
	HResult	(Inherited from Exception .)

See Also

Reference

[PipelineException Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ProcessorParameter Class

Note

This class is available only when developing for Windows.

Represents a processor parameter. Processor parameters are automatically discovered by the content pipeline. Therefore, only custom processor developers should use this class directly.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class ProcessorParameter
```

See Also

Reference

[ProcessorParameter Members](#)








[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista






ProcessorParameter Members

The following tables list the members exposed by the ProcessorParameter type.



Public Properties

Name	Description
 DefaultValue	Default value of the parameter, as specified by the [DefaultValue] attribute.
 Description	Description of the parameter, as specified by the [Description] attribute.
 DisplayName	Name of the parameter displayed in the designer, as specified by the [DisplayName] attribute.
 IsEnum	Gets a value indicating whether the parameter is an enumeration.
 PossibleEnumValues	Available options for enumerated type parameters.
 PropertyName	Name of the property, as defined in the C# code.
 PropertyType	Type of the parameter.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ProcessorParameter Class](#)



[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ProcessorParameter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[ProcessorParameter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ProcessorParameter Properties

Public Properties

	Name	Description
	DefaultValue	Default value of the parameter, as specified by the [DefaultValue] attribute.
	Description	Description of the parameter, as specified by the [Description] attribute.
	DisplayName	Name of the parameter displayed in the designer, as specified by the [DisplayName] attribute.
	IsEnum	Gets a value indicating whether the parameter is an enumeration.
	PossibleEnumValues	Available options for enumerated type parameters.
	PropertyName	Name of the property, as defined in the C# code.
	PropertyType	Type of the parameter.

See Also

Reference

[ProcessorParameter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ProcessorParameter.DefaultValue Property

Note

This property is available only when developing for Windows.

Default value of the parameter, as specified by the [DefaultValue] attribute.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object DefaultValue { get; set; }
```

Property Value

Default value of the processor parameter.

Remarks

Not all types are supported. For a complete list of supported values, see [Parameterized Processors](#).

DefaultValue is treated as a suggestion to the designer UI and is not enforced by the content pipeline. This value may be **null**.

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameter.Description Property

Note

This property is available only when developing for Windows.

Description of the parameter, as specified by the [Description] attribute.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string Description { get; set; }
```

Property Value

Parameter description.

Remarks

The description appears as a tooltip, displayed by the designer UI. This value may be **null**.

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameter.DisplayName Property

Note

This property is available only when developing for Windows.

Name of the parameter displayed in the designer, as specified by the [DisplayName] attribute.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string DisplayName { get; set; }
```

Property Value

Friendly name of the parameter.

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameter.IsEnum Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether the parameter is an enumeration.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool IsEnum { get; }
```

Property Value

true if the parameter is an enumeration; **false** otherwise.

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameter.PossibleEnumValues Property

Note

This property is available only when developing for Windows.

Available options for enumerated type parameters. For parameters of other types, this value is **null**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ReadOnlyCollection<string> PossibleEnumValues { get; }
```

Property Value

Available values for the parameter.

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameter.PropertyName Property

Note

This property is available only when developing for Windows.

Name of the property, as defined in the C# code.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string PropertyName { get; }
```

Property Value

Name of the property used to identify a specific processor parameter.

RemarksCorresponds to the property name, as defined in the C# code. This differs from the friendly parameter name, stored in the [DisplayName](#) property.

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameter.PropertyType Property

Note

This property is available only when developing for Windows.

Type of the parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string PropertyType { get; }
```

Property Value

Type of the processor parameter. For a list of accepted types, see [Parameterized Processors](#).

See Also

Reference

[ProcessorParameter Class](#)

[ProcessorParameter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ProcessorParameterCollection Class

Note

This class is available only when developing for Windows.

Represents a collection of processor [parameters](#), usually for a single processor. This class is primarily designed for internal use or for custom processor developers.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[SerializableAttribute]  
public sealed class ProcessorParameterCollection : ReadOnlyCollection<ProcessorParameter>
```

See Also

Reference

[ProcessorParameterCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista


ProcessorParameterCollection Members

The following tables list the members exposed by the ProcessorParameterCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[ProcessorParameterCollection Class](#)



[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ProcessorParameterCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also



Reference

[ProcessorParameterCollection Class](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

ProcessorParameterCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ProcessorParameterCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

VideoContent Class

Note

This class is available only when developing for Windows.

Provides a base class for all video objects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class VideoContent : ContentItem, IDisposable
```

See Also

Reference

[VideoContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista











VideoContent Members

The following tables list the members exposed by the VideoContent type.







Public Constructors

Name	Description
 VideoContent	Initializes a new copy of the VideoContent class for the specified video file.



Public Properties

Name	Description
 BitsPerSecond	Gets the bit rate for this video.
 Duration	Gets the duration of this video.
 Filename	Gets or sets the file name for this video.
 FramesPerSecond	Gets the frame rate for this video.
 Height	Gets the height of this video.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 VideoSoundtrackType	Gets or sets the type of soundtrack accompanying the video.
 Width	Gets the width of this video.

Public Methods

Name	Description
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VideoContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

VideoContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new copy of the **VideoContent** class for the specified video file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VideoContent (  
    string filename  
)
```

Parameters

filename

The file name of the video to import.

See Also

Reference

[VideoContent Class](#)







[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



Platforms Windows XP SP2, Windows Vista

VideoContent Methods

Public Methods

	Name	Description
	Dispose	Immediately releases the unmanaged resources used by this object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VideoContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

VideoContent.Dispose Method

Note

This method is available only when developing for Windows.

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[VideoContent Class](#)











[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent Properties

Public Properties

	Name	Description
	BitsPerSecond	Gets the bit rate for this video.
	Duration	Gets the duration of this video.
	Filename	Gets or sets the file name for this video.
	FramesPerSecond	Gets the frame rate for this video.
	Height	Gets the height of this video.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	VideoSoundtrackType	Gets or sets the type of soundtrack accompanying the video.
	Width	Gets the width of this video.

See Also

Reference

[VideoContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

VideoContent.BitsPerSecond Property

Note

This property is available only when developing for Windows.

Gets the bit rate for this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int BitsPerSecond { get; }
```

Property Value

The number of bits per second encoded in this video.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent.Duration Property

Note

This property is available only when developing for Windows.

Gets the duration of this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

The length of this video.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent.FileName Property

Note

This property is available only when developing for Windows.

Gets or sets the file name for this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentSerializerAttribute]  
public string Filename { get; set; }
```

Property Value

The file name for this video.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent.FramesPerSecond Property

Note

This property is available only when developing for Windows.

Gets the frame rate for this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public float FramesPerSecond { get; }
```

Property Value

The number of frames per second encoded for this video.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent.Height Property

Note

This property is available only when developing for Windows.

Gets the height of this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Height { get; }
```

Property Value

The height of this video, in pixels.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent.VideoSoundtrackType Property

Note

This property is available only when developing for Windows.

Gets or sets the type of soundtrack accompanying the video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentSerializerAttribute]  
public VideoSoundtrackType VideoSoundtrackType { get; set; }
```

Property Value

The soundtrack type.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoContent.Width Property

Note

This property is available only when developing for Windows.

Gets the width of this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Width { get; }
```

Property Value

The width of this video, in pixels.

See Also

Reference

[VideoContent Class](#)

[VideoContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista

XmlImporter Class

Note

This class is available only when developing for Windows.

Implements an importer for reading intermediate XML files. This is a wrapper around [IntermediateSerializer](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class XmlImporter : ContentImporter<Object>
```

See Also

Reference

[XmlImporter Members](#)


[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

Platforms Windows XP SP2, Windows Vista







XmlImporter Members

The following tables list the members exposed by the XmlImporter type.



Public Constructors

	Name	Description
	XmlImporter	Initializes a new instance of XmlImporter.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Import	Called by the framework when importing an intermediate file containing a game asset.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[XmlImporter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

XmlImporter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **XmlImporter**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public XmlImporter ()
```

See Also

Reference

[XmlImporter Class](#)







[XmlImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)



PlatformsWindows XP SP2, Windows Vista

XmlImporter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Import	Called by the framework when importing an intermediate file containing a game asset.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[XmlImporter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

XmlImporter.Import Method

Note

This method is available only when developing for Windows.

Called by the framework when importing an intermediate file containing a game asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override Object Import (  
    string filename,  
    ContentImporterContext context  
)
```

Parameters

filename

Name of the intermediate file.

context

Contains information for importing the intermediate file, such as a logger interface.

Return Value

Object representing the game asset contained by the intermediate file.

See Also

Reference

[XmlImporter Class](#)

[XmlImporter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline Namespace](#)

PlatformsWindows XP SP2, Windows Vista



Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace

Note




This namespace is available only when developing for Windows.

Provides intermediate classes and types for representing and manipulating graphics audio data.

Classes

	Name	Description
	AudioContent	Encapsulates and provides operations, such as format conversions, on the source audio. This type is produced by the audio importers and used by audio processors to produce compiled audio assets.
	AudioFormat	Encapsulates the native audio format (WAVEFORMATEX) information of the audio content.

Enumerations

	Name	Description
	AudioFileType	Type of the audio file.
	ConversionFormat	Target formats supported for audio source conversions.
	ConversionQuality	Compression quality of the audio content.

AudioContent Class

Note

This class is available only when developing for Windows.

Encapsulates and provides operations, such as format conversions, on the source audio. This type is produced by the audio importers and used by audio processors to produce compiled audio assets.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public class AudioContent : ContentItem, IDisposable
```

See Also

Reference

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)


PlatformsWindows XP SP2, Windows Vista

AudioContent Members











AudioContent members.

The following tables list the members exposed by the AudioContent type.








Public Constructors

Name	Description
 AudioContent	Initializes a new instance of AudioContent.



Public Properties

Name	Description
 Data	Gets the raw audio data.
 Duration	Gets the duration (in milliseconds) of the audio data.
 FileName	Gets the file name containing the audio data.
 FileType	Gets the AudioFileType of this audio source.
 Format	Gets the AudioFormat of this audio source.
 Identity	(Inherited from ContentItem .)
 LoopLength	Gets the loop length, in samples.
 LoopStart	Gets the loop start, in samples.
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)

Public Methods

Name	Description
 ConvertFormat	Transcodes the source audio to the target format and quality.
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AudioContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

AudioContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **AudioContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public AudioContent (  
    string audioFileName,  
    AudioFileType audioFileType  
)
```

Parameters

audioFileName

Name of the audio source file to be processed.

audioFileType

Type of the processed audio: WAV, MP3 or WMA.

Remarks

Constructs the object from the specified source file, in the format specified.

See Also

Reference

[AudioContent Class](#)








[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)



PlatformsWindows XP SP2, Windows Vista

AudioContent Methods

Public Methods

Name	Description
 ConvertFormat	Transcodes the source audio to the target format and quality.
 Dispose	Immediately releases the unmanaged resources used by this object.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the object.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AudioContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

AudioContent.ConvertFormat Method

Note

This method is available only when developing for Windows.

Transcodes the source audio to the target format and quality.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public void ConvertFormat (  
    ConversionFormat formatType,  
    ConversionQuality quality,  
    string targetFileName  
)
```

Parameters

formatType

Format of the processed source audio: WAV, MP3 or WMA.

quality

Quality of the processed source audio. It can be one of the following:

- Low (96 kbps)
- Medium (128 kbps)
- Best (192 kbps)

targetFileName

Name of the file containing the processed source audio.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.Dispose Method

Note

This method is available only when developing for Windows.

Immediately releases the unmanaged resources used by this object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public void Dispose ()
```

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

Platforms Windows XP SP2, Windows Vista

AudioContent.Finalize Method

Note

This method is available only when developing for Windows.

Releases unmanaged resources and performs other cleanup operations before garbage collection reclaims the object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
protected override void Finalize ()
```

See Also

Reference

[AudioContent Class](#)











[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent Properties

Public Properties

	Name	Description
	Data	Gets the raw audio data.
	Duration	Gets the duration (in milliseconds) of the audio data.
	FileName	Gets the file name containing the audio data.
	FileType	Gets the AudioFileType of this audio source.
	Format	Gets the AudioFormat of this audio source.
	Identity	(Inherited from ContentItem .)
	LoopLength	Gets the loop length, in samples.
	LoopStart	Gets the loop start, in samples.
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[AudioContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

AudioContent.Data Property

Note

This property is available only when developing for Windows.

Gets the raw audio data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public ReadOnlyCollection<byte> Data { get; }
```

Property Value

If unprocessed, the source data; otherwise, the processed data.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.Duration Property

Note

This property is available only when developing for Windows.

Gets the duration (in milliseconds) of the audio data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public TimeSpan Duration { get; }
```

Property Value

Duration of the audio data.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.FileName Property

Note

This property is available only when developing for Windows.

Gets the file name containing the audio data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
[ContentSerializerAttribute]  
public string FileName { get; }
```

Property Value

The name of the file containing this data.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.FileType Property

Note

This property is available only when developing for Windows.

Gets the [AudioFileType](#) of this audio source.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public AudioFileType FileType { get; }
```

Property Value

The [AudioFileType](#) of this audio source.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.Format Property

Note

This property is available only when developing for Windows.

Gets the [AudioFormat](#) of this audio source.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public AudioFormat Format { get; }
```

Property Value

The [AudioFormat](#) of this audio source.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.LoopLength Property

Note

This property is available only when developing for Windows.

Gets the loop length, in samples.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int LoopLength { get; }
```

Property Value

The number of samples in the loop.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioContent.LoopStart Property

Note

This property is available only when developing for Windows.

Gets the loop start, in samples.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int LoopStart { get; }
```

Property Value

The start of the loop.

See Also

Reference

[AudioContent Class](#)

[AudioContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFileType Enumeration

Note

This enumeration is available only when developing for Windows.

Type of the audio file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public enum AudioFileType
```

Members

Member name	Description
Mp3	The MP3 format
Wav	The WAV format
Wma	The WMA format

See Also

Reference

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

Platforms Windows XP SP2, Windows Vista

AudioFormat Class

Note

This class is available only when developing for Windows.

Encapsulates the native audio format (WAVEFORMATEX) information of the audio content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public sealed class AudioFormat
```

See Also

Reference

[AudioFormat Members](#)








[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

Platforms Windows XP SP2, Windows Vista






AudioFormat Members

The following tables list the members exposed by the AudioFormat type.



Public Properties

Name	Description
 AverageBytesPerSecond	Gets the average bytes processed per second.
 BitsPerSample	Gets the bit depth of the audio content.
 BlockAlign	Gets the number of bytes per sample block, taking channels into consideration.
 ChannelCount	Gets the number of channels.
 Format	Gets the format of the audio content.
 NativeWaveFormat	Gets the raw byte buffer for the format.
 SampleRate	Gets the sample rate of the audio content.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[AudioFormat Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

AudioFormat Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[AudioFormat Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

AudioFormat Properties

Public Properties

	Name	Description
	AverageBytesPerSecond	Gets the average bytes processed per second.
	BitsPerSample	Gets the bit depth of the audio content.
	BlockAlign	Gets the number of bytes per sample block, taking channels into consideration.
	ChannelCount	Gets the number of channels.
	Format	Gets the format of the audio content.
	NativeWaveFormat	Gets the raw byte buffer for the format.
	SampleRate	Gets the sample rate of the audio content.

See Also

Reference

[AudioFormat Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

AudioFormat.AverageBytesPerSecond Property

Note

This property is available only when developing for Windows.

Gets the average bytes processed per second.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int AverageBytesPerSecond { get; }
```

Property Value

Average bytes processed per second.

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFormat.BitsPerSample Property

Note

This property is available only when developing for Windows.

Gets the bit depth of the audio content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int BitsPerSample { get; }
```

Property Value

If the audio has not been processed, the source bit depth; otherwise, the bit depth of the new format.

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFormat.BlockAlign Property

Note

This property is available only when developing for Windows.

Gets the number of bytes per sample block, taking channels into consideration. For example, for 16-bit stereo audio (PCM format), the size of each sample block is 4 bytes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int BlockAlign { get; }
```

Property Value

Number of bytes, per sample block.

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFormat.ChannelCount Property

Note

This property is available only when developing for Windows.

Gets the number of channels.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int ChannelCount { get; }
```

Property Value

If the audio has not been processed, the source channel count; otherwise, the new channel count.

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFormat.Format Property

Note

This property is available only when developing for Windows.

Gets the format of the audio content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int Format { get; }
```

Property Value

If the audio has not been processed, the format tag of the source content; otherwise, the new format tag.

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFormat.NativeWaveFormat Property

Note

This property is available only when developing for Windows.

Gets the raw byte buffer for the format. For non-PCM formats, this buffer contains important format specific information beyond the basic format information exposed in other properties of the [AudioFormat](#) type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public ReadOnlyCollection<byte> NativeWaveFormat { get; }
```

Property Value

TBD

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AudioFormat.SampleRate Property

Note

This property is available only when developing for Windows.

Gets the sample rate of the audio content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public int SampleRate { get; }
```

Property Value

If the audio has not been processed, the source sample rate; otherwise, the new sample rate.

See Also

Reference

[AudioFormat Class](#)

[AudioFormat Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ConversionFormat Enumeration

Note

This enumeration is available only when developing for Windows.

Target formats supported for audio source conversions.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public enum ConversionFormat
```

Members

Member name	Description
Adpcm	A PCM encoding technique using 4 bits
Pcm	8/16-bit mono/stereo PCM audio 8KHz-48KHz
WindowsMedia	Windows Media CBR formats (64 kbps, 128 kbps, 192 kbps)
Xma	The Xbox compression format

See Also

Reference

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

Platforms Windows XP SP2, Windows Vista

ConversionQuality Enumeration

Note

This enumeration is available only when developing for Windows.

Compression quality of the audio content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Audio

Assembly: (in)

Syntax

C#

```
public enum ConversionQuality
```

Members

Member name	Description
Low	High compression yielding lower file size, but could compromise audio quality
Medium	Moderate compression resulting in a compromise between audio quality and file size
Best	Lowest compression, but the best audio quality

See Also

Reference

[Microsoft.Xna.Framework.Content.Pipeline.Audio Namespace](#)

Platforms Windows XP SP2, Windows Vista

Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace














Note

This namespace is available only when developing for Windows.


Provides intermediate classes and types for representing and manipulating graphics data.

Classes


Name	Description
AnimationChannel	Provides methods and properties for maintaining an animation channel.
AnimationChannelDictionary	Collection of animation data channels, one per bone or rigid object.
AnimationContent	Provides properties for maintaining an animation.
AnimationContentDictionary	Collection of named animations.
AnimationKeyframe	Provides methods and properties for managing a keyframe.
BasicMaterialContent	Provides properties for modifying a traditional fixed-function-style material, as supported by most 3D modeling packages.
BitmapContent	Provides properties and methods for creating and maintaining a bitmap resource.
BoneContent	Represents an animation skeleton.
BoneWeightCollection	Collection of bone weights of a vertex.
Dxt1BitmapContent	Provides methods and properties for managing compressed textures (DXT1).
Dxt3BitmapContent	Provides methods and properties for managing compressed textures (DXT3).
Dxt5BitmapContent	Provides methods and properties for managing compressed textures (DXT5).
DxtBitmapContent	Provides methods and properties for managing compressed textures (DXT1, DXT3, DXT5).
EffectContent	Contains the source code for a DirectX Effect, loaded from a .fx file.
EffectMaterialContent	Provides support for representing DirectX Effect materials.
FontDescription	Provides information to the FontDescriptionProcessor describing which font to rasterize, which font size to utilize, and which Unicode characters to include in the processor output.
GeometryContent	Provides properties that define various aspects of a geometry batch.
GeometryContentCollection	Provides methods for maintaining a collection of geometry batches that make up a mesh.
IndexCollection	Provides methods for maintaining a list of index values.
IndirectPositionCollection	Provides methods for maintaining a list of vertex positions.
MaterialContent	Provides methods and properties for maintaining a collection of named texture references.
MeshBuilder	Provides support for writing a custom importer for mesh objects.
MeshContent	Provides properties and methods that define various aspects of a mesh.
MeshHelper	Provides methods for manipulating mesh data.
MipmapChain	Provides methods for accessing a mipmap chain.
MipmapChainCollection	Provides methods for maintaining a mipmap chain.
NodeContent	Provides a base class for graphics types that define local coordinate systems.
NodeContentCollection	Provides a collection of all NodeContent objects in a spatial hierarchy.

 PixelBitmapContent	Provides methods for maintaining a 2D array of pixel values.
 PositionCollection	Provides a collection of vertex position values.
 Texture2DContent	Represents a regular two-dimensional texture.
 Texture3DContent	Represents a three-dimensional volume texture.
 TextureContent	Provides a base class for all texture objects.
 TextureCubeContent	Provides validation for a cube map texture, which contains an array of six image faces.
 TextureReferenceDictionary	Provides a collection of named references to texture files.
 VectorConverter	Provides methods for converting data between different packed vector representations.
 VertexChannel	Provides methods and properties for maintaining a vertex channel.
 VertexChannel	Provides methods and properties for maintaining a vertex channel.
 VertexChannelCollection	Provides methods and properties for managing a list of vertex data channels.
 VertexChannelNames	Provides properties for managing a collection of vertex channel names.
 VertexContent	Provides methods and properties for maintaining the vertex data of a GeometryContent .

Structures

Name	Description
 BoneWeight	Provides properties for managing a bone weight.

Enumerations

Name	Description
 FontDescriptionStyle	Flags that describe style information to be applied to text.

AnimationChannel Class

Note

This class is available only when developing for Windows.

Provides methods and properties for maintaining an animation channel. An animation channel is a collection of keyframes describing the movement of a single bone or rigid object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class AnimationChannel : ICollection<AnimationKeyframe>, IEnumerable<AnimationKeyframe>, IEnumerable
```

See Also

Reference

[AnimationChannel Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



AnimationChannel Members

The following tables list the members exposed by the AnimationChannel type.













Public Constructors

Name	Description
 AnimationChannel	Initializes a new instance of AnimationChannel.



Public Properties

Name	Description
 Count	Gets the number of keyframes in the collection.
 Item	Gets the keyframe at the specified index position.





Public Methods

Name	Description
 Add	Adds a new keyframe to the collection, automatically sorting the contents according to keyframe times.
 Clear	Removes all keyframes from the collection.
 Contains	Searches the collection for the specified keyframe.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the keyframes.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Determines the index for the specified keyframe.
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the specified keyframe from the collection.
 RemoveAt	Removes the keyframe at the specified index position.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.IsReadOnly	Returns a value indicating whether the object is read-only.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.Add	Adds a new keyframe to the collection, automatically sorting the contents according to keyframe times.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.CopyTo	Copies the elements of the animation channel to an array, starting at the specified index.
 System.Collections.IEnumerable.GetEnumerator	Gets an enumerator that iterates through the keyframes of an animation channel.

See Also

Reference

[AnimationChannel Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationChannel Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **AnimationChannel**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationChannel ()
```

See Also

Reference

[AnimationChannel Class](#)













[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

AnimationChannel Methods





Public Methods

Name	Description
 Add	Adds a new keyframe to the collection, automatically sorting the contents according to keyframe times.
 Clear	Removes all keyframes from the collection.
 Contains	Searches the collection for the specified keyframe.
 Equals	(Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the keyframes.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Determines the index for the specified keyframe.
 ReferenceEquals	(Inherited from Object .)
 Remove	Removes the specified keyframe from the collection.
 RemoveAt	Removes the keyframe at the specified index position.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.IsReadOnly	Returns a value indicating whether the object is read-only.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.Add	Adds a new keyframe to the collection, automatically sorting the contents according to keyframe times.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.CopyTo	Copies the elements of the animation channel to an array, starting at the specified index.
 System.Collections.IEnumerable.GetEnumerator	Gets an enumerator that iterates through the keyframes of an animation channel.

See Also

Reference

[AnimationChannel Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationChannel.Add Method

Note

This method is available only when developing for Windows.

Adds a new keyframe to the collection, automatically sorting the contents according to keyframe times.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Add (  
    AnimationKeyframe item  
)
```

Parameters

item

Keyframe to be added to the channel.

Return Value

Index of the new keyframe.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

AnimationChannel.Clear Method

Note

This method is available only when developing for Windows.

Removes all keyframes from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Clear ()
```

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel.Contains Method

Note

This method is available only when developing for Windows.

Searches the collection for the specified keyframe.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Contains (  
    AnimationKeyframe item  
)
```

Parameters

item

Keyframe being searched for.

Return Value

true if the keyframe exists; **false** otherwise.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

AnimationChannel.GetEnumerator Method

Note

This method is available only when developing for Windows.

Returns an enumerator that iterates through the keyframes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerator<AnimationKeyframe> GetEnumerator ()
```

Return Value

Enumerator for the keyframe collection.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel.IndexOf Method

Note

This method is available only when developing for Windows.

Determines the index for the specified keyframe.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int IndexOf (  
    AnimationKeyframe item  
)
```

Parameters

item

Identity of a keyframe.

Return Value

Index of the specified keyframe.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel.Remove Method

Note

This method is available only when developing for Windows.

Removes the specified keyframe from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Remove (  
    AnimationKeyframe item  
)
```

Parameters

item

Keyframe being removed.

Return Value

true if the keyframe was removed; **false** otherwise.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel.RemoveAt Method

Note

This method is available only when developing for Windows.

Removes the keyframe at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void RemoveAt (  
    int index  
)
```

Parameters

index

Index of the keyframe being removed.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.Add Method

Note

This method is available only when developing for Windows.

Adds a new keyframe to the collection, automatically sorting the contents according to keyframe times.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.Add (
    AnimationKeyframe item
)
```

Parameters

item

Keyframe to be added to the channel.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.CopyTo Method

Note

This method is available only when developing for Windows.

Copies the elements of the animation channel to an array, starting at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.CopyTo (
    AnimationKeyframe[] array,
    int arrayIndex
)
```

Parameters

array

Array that will receive the copied animation channel elements.

arrayIndex

Starting index for copy operation.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.IEnumerable.GetEnumerator Method

Note

This method is available only when developing for Windows.

Gets an enumerator that iterates through the keyframes of an animation channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

An enumerator for the [AnimationChannel](#).

See Also

Reference

[AnimationChannel Class](#)



[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel Properties

Public Properties

	Name	Description
	Count	Gets the number of keyframes in the collection.
	Item	Gets the keyframe at the specified index position.

See Also

Reference

[AnimationChannel Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationChannel.Count Property

Note

This property is available only when developing for Windows.

Gets the number of keyframes in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of keyframes.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel.Item Property

Note

This property is available only when developing for Windows.

Gets the keyframe at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationKeyframe this [
    int index
] { get; }
```

Property Value

Retrieved keyframe.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannel.System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline> Property

Note

This property is available only when developing for Windows.

Returns a value indicating whether the object is read-only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.AnimationKeyframe>.IsReadOnly { get; }
```

Property Value

true if this object is read-only; **false** otherwise.

See Also

Reference

[AnimationChannel Class](#)

[AnimationChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationChannelDictionary Class

Note

This class is available only when developing for Windows.

Collection of animation data channels, one per bone or rigid object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class AnimationChannelDictionary : NamedValueDictionary<AnimationChannel>
```

Remarks

This collection is based on the name of the object being animated by each channel. Typically, target objects are an instance of [BoneContent](#), but they can be any [NodeContent](#) subclass.

See Also

Reference

[AnimationChannelDictionary Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista





AnimationChannelDictionary Members

The following tables list the members exposed by the AnimationChannelDictionary type.












Public Constructors

Name	Description
 AnimationChannelDictionary	Initializes a new instance of AnimationChannelDictionary.







Public Properties

Name	Description
 Count	(Inherited from NamedValueDictionary .)
 Item	(Inherited from NamedValueDictionary .)
 Keys	(Inherited from NamedValueDictionary .)
 Values	(Inherited from NamedValueDictionary .)

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 AddItem	(Inherited from NamedValueDictionary .)
 ClearItems	(Inherited from NamedValueDictionary .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from NamedValueDictionary .)
 SetItem	(Inherited from NamedValueDictionary .)

See Also

Reference

[AnimationChannelDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationChannelDictionary Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **AnimationChannelDictionary**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationChannelDictionary ()
```

See Also

Reference

[AnimationChannelDictionary Class](#)












[AnimationChannelDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)







PlatformsWindows XP SP2, Windows Vista

AnimationChannelDictionary Methods

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 AddItem	(Inherited from NamedValueDictionary .)
 ClearItems	(Inherited from NamedValueDictionary .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from NamedValueDictionary .)
 SetItem	(Inherited from NamedValueDictionary .)

See Also





Reference

[AnimationChannelDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationChannelDictionary Properties

Public Properties

	Name	Description
	Count	(Inherited from NamedValueDictionary.)
	Item	(Inherited from NamedValueDictionary.)
	Keys	(Inherited from NamedValueDictionary.)
	Values	(Inherited from NamedValueDictionary.)

See Also

Reference

[AnimationChannelDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationContent Class

Note

This class is available only when developing for Windows.

Provides properties for maintaining an animation.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class AnimationContent : ContentItem
```

Remarks

An animation contains a collection of data channels, describing a complete set of movement for any number of bones or rigid objects. The channels are stored in the [Animations](#) dictionary. For character skinning, the animation data is usually attached to the root of a bone hierarchy. However, it can potentially belong to any node. For example, when used for rigid body animation.

See Also

Reference

[AnimationContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista






AnimationContent Members

The following tables list the members exposed by the AnimationContent type.






Public Constructors

	Name	Description
	AnimationContent	Initializes a new instance of AnimationContent.



Public Properties

	Name	Description
	Channels	Gets the collection of animation data channels.
	Duration	Gets or sets the total length of the animation.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AnimationContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **AnimationContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationContent ()
```

See Also

Reference

[AnimationContent Class](#)






[AnimationContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

AnimationContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[AnimationContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationContent Properties

Public Properties

	Name	Description
	Channels	Gets the collection of animation data channels.
	Duration	Gets or sets the total length of the animation.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[AnimationContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationContent.Channels Property

Note

This property is available only when developing for Windows.

Gets the collection of animation data channels. Each channel describes the movement of a single bone or rigid object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationChannelDictionary Channels { get; }
```

Property Value

Collection of animation channels.

See Also

Reference

[AnimationContent Class](#)

[AnimationContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationContent.Duration Property

Note

This property is available only when developing for Windows.

Gets or sets the total length of the animation.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TimeSpan Duration { get; set; }
```

Property Value

Length of animation.

See Also

Reference

[AnimationContent Class](#)

[AnimationContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationContentDictionary Class

Note

This class is available only when developing for Windows.

Collection of named animations.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class AnimationContentDictionary : NamedValueDictionary<AnimationContent>
```

See Also

Reference

[AnimationContentDictionary Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista





AnimationContentDictionary Members

The following tables list the members exposed by the AnimationContentDictionary type.












Public Constructors

Name	Description
 AnimationContentDictionary	Initializes a new instance of AnimationContentDictionary.







Public Properties

Name	Description
 Count	(Inherited from NamedValueDictionary .)
 Item	(Inherited from NamedValueDictionary .)
 Keys	(Inherited from NamedValueDictionary .)
 Values	(Inherited from NamedValueDictionary .)

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 AddItem	(Inherited from NamedValueDictionary .)
 ClearItems	(Inherited from NamedValueDictionary .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from NamedValueDictionary .)
 SetItem	(Inherited from NamedValueDictionary .)

See Also

Reference

[AnimationContentDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationContentDictionary Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **AnimationContentDictionary**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationContentDictionary ()
```

See Also

Reference

[AnimationContentDictionary Class](#)












[AnimationContentDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)







PlatformsWindows XP SP2, Windows Vista

AnimationContentDictionary Methods

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 AddItem	(Inherited from NamedValueDictionary .)
 ClearItems	(Inherited from NamedValueDictionary .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from NamedValueDictionary .)
 SetItem	(Inherited from NamedValueDictionary .)

See Also





Reference

[AnimationContentDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationContentDictionary Properties

Public Properties

	Name	Description
	Count	(Inherited from NamedValueDictionary.)
	Item	(Inherited from NamedValueDictionary.)
	Keys	(Inherited from NamedValueDictionary.)
	Values	(Inherited from NamedValueDictionary.)

See Also

Reference

[AnimationContentDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationKeyframe Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing a keyframe. A keyframe describes the position of an animation channel at a single point in time.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class AnimationKeyframe : IComparable<AnimationKeyframe>
```

See Also

Reference

[AnimationKeyframe Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista



AnimationKeyframe Members

The following tables list the members exposed by the AnimationKeyframe type.







Public Constructors

Name	Description
 AnimationKeyframe	Initializes a new instance of AnimationKeyframe with the specified time offset and transform.



Public Properties

Name	Description
 Time	Gets the time offset from the start of the animation to the position described by this keyframe.
 Transform	Gets or sets the position described by this keyframe.

Public Methods

Name	Description
 CompareTo	Compares this instance of a keyframe to another.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AnimationKeyframe Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationKeyframe Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **AnimationKeyframe** with the specified time offset and transform.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationKeyframe (  
    TimeSpan time,  
    Matrix transform  
)
```

Parameters

time

Time offset of the keyframe.

transform

Position of the keyframe.

See Also

Reference

[AnimationKeyframe Class](#)







[AnimationKeyframe Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



Platforms Windows XP SP2, Windows Vista

AnimationKeyframe Methods

Public Methods

	Name	Description
	CompareTo	Compares this instance of a keyframe to another.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[AnimationKeyframe Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationKeyframe.CompareTo Method

Note

This method is available only when developing for Windows.

Compares this instance of a keyframe to another.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int CompareTo (  
    AnimationKeyframe other  
)
```

Parameters

other

Keyframe being compared to.

Return Value

Indication of their relative values.

See Also

Reference

[AnimationKeyframe Class](#)



[AnimationKeyframe Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

AnimationKeyframe Properties

Public Properties

	Name	Description
	Time	Gets the time offset from the start of the animation to the position described by this keyframe.
	Transform	Gets or sets the position described by this keyframe.

See Also

Reference

[AnimationKeyframe Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

AnimationKeyframe.Time Property

Note

This property is available only when developing for Windows.

Gets the time offset from the start of the animation to the position described by this keyframe.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TimeSpan Time { get; }
```

Property Value

Offset from the animation start time.

See Also

Reference

[AnimationKeyframe Class](#)

[AnimationKeyframe Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

AnimationKeyframe.Transform Property

Note

This property is available only when developing for Windows.

Gets or sets the position described by this keyframe.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Matrix Transform { get; set; }
```

Property Value

Position of keyframe.

See Also

Reference

[AnimationKeyframe Class](#)

[AnimationKeyframe Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent Class

Note

This class is available only when developing for Windows.

Provides properties for modifying a traditional fixed-function–style material, as supported by most 3D modeling packages.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class BasicMaterialContent : MaterialContent
```

See Also

Reference

[BasicMaterialContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista








BasicMaterialContent Members

The following tables list the members exposed by the BasicMaterialContent type.












Public Constructors

Name	Description
 BasicMaterialContent	Initializes a new instance of BasicMaterialContent.






Public Fields

Name	Description
 AlphaKey	Specifies the key of the key/data pair for the alpha property.
 DiffuseColorKey	Specifies the key of the key/data pair for the diffuse color.
 EmissiveColorKey	Specifies the key of the key/data pair for the emissive color.
 SpecularColorKey	Specifies the key of the key/data pair for the specular color.
 SpecularPowerKey	Specifies the key of the key/data pair for the specular power.
 TextureKey	Specifies the key of the key/data pair for the diffuse texture.
 VertexColorEnabledKey	Specifies the key of the key/data pair for the vertex color property.








Public Properties

Name	Description
 Alpha	Gets or sets the alpha property.
 DiffuseColor	Gets or sets the diffuse color property.
 EmissiveColor	Gets or sets the emissive color property.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 SpecularColor	Gets or sets the specular color property.
 SpecularPower	Gets or sets the specular power property.
 Texture	Gets or sets the diffuse texture property.
 Textures	(Inherited from MaterialContent .)
 VertexColorEnabled	Gets or sets the vertex color property.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetReferenceTypeProperty	(Inherited from MaterialContent .)
 GetTexture	(Inherited from MaterialContent .)
 GetValueTypeProperty	(Inherited from MaterialContent .)
 MemberwiseClone	(Inherited from Object .)
 SetProperty	(Inherited from MaterialContent .)
 SetTexture	(Inherited from MaterialContent .)

See Also








Reference

[BasicMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BasicMaterialContent Fields

Public Fields

	Name	Description
	AlphaKey	Specifies the key of the key/data pair for the alpha property.
	DiffuseColorKey	Specifies the key of the key/data pair for the diffuse color.
	EmissiveColorKey	Specifies the key of the key/data pair for the emissive color.
	SpecularColorKey	Specifies the key of the key/data pair for the specular color.
	SpecularPowerKey	Specifies the key of the key/data pair for the specular power.
	TextureKey	Specifies the key of the key/data pair for the diffuse texture.
	VertexColorEnabledKey	Specifies the key of the key/data pair for the vertex color property.

See Also

Reference

[BasicMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BasicMaterialContent.AlphaKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the alpha property.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string AlphaKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.DiffuseColorKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the diffuse color.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string DiffuseColorKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.EmissiveColorKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the emissive color.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string EmissiveColorKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.SpecularColorKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the specular color.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string SpecularColorKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.SpecularPowerKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the specular power.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string SpecularPowerKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.TextureKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the diffuse texture.

This key/data pair is stored in the [Textures](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string TextureKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.VertexColorEnabledKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the vertex color property.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string VertexColorEnabledKey
```

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **BasicMaterialContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BasicMaterialContent ()
```

See Also

Reference

[BasicMaterialContent Class](#)






[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)








PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	GetReferenceTypeProperty	(Inherited from MaterialContent .)
	GetTexture	(Inherited from MaterialContent .)
	GetValueTypeProperty	(Inherited from MaterialContent .)
	MemberwiseClone	(Inherited from Object .)
	SetProperty	(Inherited from MaterialContent .)
	SetTexture	(Inherited from MaterialContent .)

See Also












Reference

[BasicMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BasicMaterialContent Properties

Public Properties

	Name	Description
	Alpha	Gets or sets the alpha property.
	DiffuseColor	Gets or sets the diffuse color property.
	EmissiveColor	Gets or sets the emissive color property.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	SpecularColor	Gets or sets the specular color property.
	SpecularPower	Gets or sets the specular power property.
	Texture	Gets or sets the diffuse texture property.
	Textures	(Inherited from MaterialContent .)
	VertexColorEnabled	Gets or sets the vertex color property.

See Also

Reference

[BasicMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BasicMaterialContent.Alpha Property

Note

This property is available only when developing for Windows.

Gets or sets the alpha property.

This is a shortcut for accessing **Textures[AlphaKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Nullable<float> Alpha { get; set; }
```

Property Value

Current alpha value or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.DiffuseColor Property

Note

This property is available only when developing for Windows.

Gets or sets the diffuse color property.

This is a shortcut for accessing **OpaqueData[DiffuseColorKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Nullable<Vector3> DiffuseColor { get; set; }
```

Property Value

Current diffuse color value or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.EmissiveColor Property

Note

This property is available only when developing for Windows.

Gets or sets the emissive color property.

This is a shortcut for accessing **OpaqueData[EmissiveColorKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Nullable<Vector3> EmissiveColor { get; set; }
```

Property Value

Current diffuse color value or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.SpecularColor Property

Note

This property is available only when developing for Windows.

Gets or sets the specular color property.

This is a shortcut for accessing **OpaqueData[SpecularColorKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Nullable<Vector3> SpecularColor { get; set; }
```

Property Value

Current specular color value or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.SpecularPower Property

Note

This property is available only when developing for Windows.

Gets or sets the specular power property.

This is a shortcut for accessing **OpaqueData[SpecularPowerKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Nullable<float> SpecularPower { get; set; }
```

Property Value

Current specular power value or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.Texture Property

Note

This property is available only when developing for Windows.

Gets or sets the diffuse texture property.

This is a shortcut for accessing **Textures[DiffuseTextureKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference<TextureContent> Texture { get; set; }
```

Property Value

Current diffuse texture value or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BasicMaterialContent.VertexColorEnabled Property

Note

This property is available only when developing for Windows.

Gets or sets the vertex color property.

This is a shortcut for accessing **Textures[VertexColorEnabledKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Nullable<bool> VertexColorEnabled { get; set; }
```

Property Value

Current vertex color or the value to be set.

See Also

Reference

[BasicMaterialContent Class](#)

[BasicMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent Class

Note

This class is available only when developing for Windows.

Provides properties and methods for creating and maintaining a bitmap resource.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class BitmapContent : ContentItem
```

Remarks

The **BitmapContent** class represents a single two-dimensional image with a fixed size and fixed format. Various pixel encodings are supported by subclasses of the bitmap type. Higher level constructs such as textures that may contain multiple mipmap images or cube map faces are stored as collections of multiple bitmaps.

See Also

Reference

[BitmapContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista






BitmapContent Members

The following tables list the members exposed by the BitmapContent type.











Public Constructors

Name	Description
 BitmapContent	Overloaded. Initializes a new instance of BitmapContent .







Public Properties

Name	Description
 Height	Gets or sets the height of the bitmap, in pixels.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Width	Gets or sets the width of the bitmap, in pixels.

Public Methods

Name	Description
  Copy	Overloaded. Copies one bitmap to another.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetPixelData	Reads encoded bitmap content.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetPixelData	Writes encoded bitmap content.
 ToString	Returns a string description of the bitmap resource.
 TryGetFormat	Gets the corresponding GPU texture format for the specified bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 TryCopyFrom	Attempts to copy a region from a specified bitmap.
 TryCopyTo	Attempts to copy a region of the specified bitmap onto another.
  ValidateCopyArguments	Validates the arguments to the Copy function.

See Also

Reference

[BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BitmapContent Constructor

Initializes a new instance of [BitmapContent](#).

Overload List

Name	Description
BitmapContent ()	Initializes a new instance of BitmapContent .
BitmapContent (Int32, Int32)	Initializes a new instance of BitmapContent with the specified width or height.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BitmapContent Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [BitmapContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected BitmapContent ()
```

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent Constructor (Int32, Int32)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [BitmapContent](#) with the specified width or height.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected BitmapContent (  
    int width,  
    int height  
)
```

Parameters

width

Width, in pixels, of the bitmap resource.

height

Height, in pixels, of the bitmap resource.

See Also

Reference

[BitmapContent Class](#)










[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)






PlatformsWindows XP SP2, Windows Vista

BitmapContent Methods

Public Methods

	Name	Description
	C Copy	Overloaded. Copies one bitmap to another.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetPixelData	Reads encoded bitmap content.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetPixelData	Writes encoded bitmap content.
	ToString	Returns a string description of the bitmap resource.
	TryGetFormat	Gets the corresponding GPU texture format for the specified bitmap type.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	TryCopyFrom	Attempts to copy a region from a specified bitmap.
	TryCopyTo	Attempts to copy a region of the specified bitmap onto another.
	S ValidateCopyArguments	Validates the arguments to the Copy function.

See Also

Reference

[BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BitmapContent.Copy Method

Copies one bitmap to another.

The destination bitmap can be in any format and size. If the destination is larger or smaller, the source bitmap is scaled accordingly.

Overload List

Name	Description
BitmapContent.Copy (BitmapContent, BitmapContent)	Copies one bitmap into another.
BitmapContent.Copy (BitmapContent, Rectangle, BitmapContent, Rectangle)	Copies one bitmap into another.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BitmapContent.Copy Method (BitmapContent, BitmapContent)

Note

This method is available only when developing for Windows.

Copies one bitmap into another.

The destination bitmap can be in any format and size. If the destination is larger or smaller, the source bitmap is scaled accordingly.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void Copy (  
    BitmapContent sourceBitmap,  
    BitmapContent destinationBitmap  
)
```

Parameters

sourceBitmap

[BitmapContent](#) being copied.

destinationBitmap

[BitmapContent](#) being overwritten.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.Copy Method (BitmapContent, Rectangle, BitmapContent, Rectangle)

Note

This method is available only when developing for Windows.

Copies one bitmap into another.

The destination bitmap can be in any format and size. If the destination is larger or smaller, the source bitmap is scaled accordingly.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void Copy (  
    BitmapContent sourceBitmap,  
    Rectangle sourceRegion,  
    BitmapContent destinationBitmap,  
    Rectangle destinationRegion  
)
```

Parameters

sourceBitmap

[BitmapContent](#) being copied.

sourceRegion

Region of *sourceBitmap*.

destinationBitmap

[BitmapContent](#) being overwritten.

destinationRegion

Region of bitmap to be overwritten.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.GetPixelData Method

Note

This method is available only when developing for Windows.

Reads encoded bitmap content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract byte[] GetPixelData ()
```

Return Value

Array containing encoded bitmap data.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.SetPixelData Method

Note

This method is available only when developing for Windows.

Writes encoded bitmap content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract void SetPixelData (  
    byte[] sourceData  
)
```

Parameters

sourceData

Array containing encoded bitmap data to be set.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.ToString Method

Note

This method is available only when developing for Windows.

Returns a string description of the bitmap resource.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

Description of the bitmap.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.TryCopyFrom Method

Note

This method is available only when developing for Windows.

Attempts to copy a region from a specified bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected abstract bool TryCopyFrom (  
    BitmapContent sourceBitmap,  
    Rectangle sourceRegion,  
    Rectangle destinationRegion  
)
```

Parameters

sourceBitmap

[BitmapContent](#) being copied.

sourceRegion

Location of *sourceBitmap*.

destinationRegion

Region of destination bitmap to be overwritten.

Return Value

true if region copy is supported; **false** otherwise.

Remarks

This operation succeeds if the source is a [PixelBitmapContent](#) object and the copy region covers all of both bitmaps. Some bitmap subclasses may support subregion copies and/or a wider range of formats than just [Vector4](#), but this is not required. Copy functionality is accessible using the [Copy](#) methods.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.TryCopyTo Method

Note

This method is available only when developing for Windows.

Attempts to copy a region of the specified bitmap onto another.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected abstract bool TryCopyTo (  
    BitmapContent destinationBitmap,  
    Rectangle sourceRegion,  
    Rectangle destinationRegion  
)
```

Parameters

destinationBitmap

[BitmapContent](#) being overwritten.

sourceRegion

Location of the source bitmap.

destinationRegion

Region of destination bitmap to be overwritten.

Return Value

true if region copy is supported; **false** otherwise.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

BitmapContent.TryGetFormat Method

Note

This method is available only when developing for Windows.

Gets the corresponding GPU texture format for the specified bitmap type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract bool TryGetFormat (  
    out SurfaceFormat format  
)
```

Parameters

format

[[OutAttribute](#)] Format being compared.

Return Value

true if the bitmap matches *format*; **false** otherwise.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.ValidateCopyArguments Method

Note

This method is available only when developing for Windows.

Validates the arguments to the [Copy](#) function.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected static void ValidateCopyArguments (  
    BitmapContent sourceBitmap,  
    Rectangle sourceRegion,  
    BitmapContent destinationBitmap,  
    Rectangle destinationRegion  
)
```

Parameters

sourceBitmap

[BitmapContent](#) being copied.

sourceRegion

Location of *sourceBitmap*.

destinationBitmap

[BitmapContent](#) being overwritten.

destinationRegion

Region of bitmap to be overwritten.

Remarks

Override this method to verify arguments from custom implementations of [TryCopyFrom](#) and [TryCopyTo](#) for your derived type.

See Also

Reference

[BitmapContent Class](#)






[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent Properties

Public Properties

	Name	Description
	Height	Gets or sets the height of the bitmap, in pixels.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Width	Gets or sets the width of the bitmap, in pixels.

See Also

Reference

[BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BitmapContent.Height Property

Note

This property is available only when developing for Windows.

Gets or sets the height of the bitmap, in pixels.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Height { get; set; }
```

Property Value

Height of the bitmap.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BitmapContent.Width Property

Note

This property is available only when developing for Windows.

Gets or sets the width of the bitmap, in pixels.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Width { get; set; }
```

Property Value

Width of the bitmap.

See Also

Reference

[BitmapContent Class](#)

[BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoneContent Class

Note

This class is available only when developing for Windows.

Represents an animation skeleton.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class BoneContent : NodeContent
```

Remarks

Animation skeletons are represented as a tree of **BoneContent** objects with the bind pose stored in their [Transform](#) properties. Animation data for the entire skeleton is stored in the [Animations](#) property of the root bone (represented by a **BoneContent** object); typically a sibling of the mesh being skinned.

See Also

Reference

[BoneContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista









BoneContent Members

The following tables list the members exposed by the BoneContent type.






Public Constructors

	Name	Description
	BoneContent	Initializes a new instance of BoneContent.



Public Properties

	Name	Description
	AbsoluteTransform	(Inherited from NodeContent .)
	Animations	(Inherited from NodeContent .)
	Children	(Inherited from NodeContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Parent	(Inherited from NodeContent .)
	Transform	(Inherited from NodeContent .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BoneContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **BoneContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BoneContent ()
```

See Also

Reference

[BoneContent Class](#)






[BoneContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

BoneContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[BoneContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneContent Properties

Public Properties

	Name	Description
	AbsoluteTransform	(Inherited from NodeContent .)
	Animations	(Inherited from NodeContent .)
	Children	(Inherited from NodeContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Parent	(Inherited from NodeContent .)
	Transform	(Inherited from NodeContent .)

See Also

Reference

[BoneContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeight Structure

Note

This structure is available only when developing for Windows.

Provides properties for managing a bone weight.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public struct BoneWeight
```

Remarks

A bone weight describes the influence of a single bone on the position of a vertex. These are grouped into [BoneWeightCollection](#) objects, which are stored in a vertex data channel.

See Also

Reference

[BoneWeight Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista



BoneWeight Members

The following tables list the members exposed by the BoneWeight type.




Public Constructors

Name	Description
 BoneWeight	Initializes a new instance of BoneWeight with the specified name and weight.



Public Properties

Name	Description
 BoneName	Gets the name of the bone.
 Weight	Gets the amount of bone influence, ranging from zero to one.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BoneWeight Structure](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeight Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **BoneWeight** with the specified name and weight.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BoneWeight (  
    string boneName,  
    float weight  
)
```

Parameters

boneName

Name of the bone.

weight

Amount of influence, ranging from zero to one.

See Also

Reference

[BoneWeight Structure](#)




[BoneWeight Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



Platforms Windows XP SP2, Windows Vista

BoneWeight Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also



Reference

[BoneWeight Structure](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeight Properties

Public Properties

	Name	Description
	BoneName	Gets the name of the bone.
	Weight	Gets the amount of bone influence, ranging from zero to one.

See Also

Reference

[BoneWeight Structure](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeight.BoneName Property

Note

This property is available only when developing for Windows.

Gets the name of the bone.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string BoneName { get; }
```

Property Value

Name of the bone.

Remarks

The bone can be located by searching for a [BoneContent](#) object with a matching [Name](#) property.

See Also

Reference

[BoneWeight Structure](#)

[BoneWeight Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoneWeight.Weight Property

Note

This property is available only when developing for Windows.

Gets the amount of bone influence, ranging from zero to one. The complete set of weights in a [BoneWeightCollection](#) should sum to one.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public float Weight { get; }
```

Property Value

Influence of the bone.

See Also

Reference

[BoneWeight Structure](#)

[BoneWeight Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoneWeightCollection Class

Note

This class is available only when developing for Windows.

Collection of bone weights of a vertex.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class BoneWeightCollection : Collection<BoneWeight>
```

Remarks

These objects are stored in the [Weights](#) property of a vertex data channel object. The complete set of weights should sum to one.

See Also

Reference

[BoneWeightCollection Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



BoneWeightCollection Members

The following tables list the members exposed by the BoneWeightCollection type.


Public Constructors

Name	Description
 BoneWeightCollection	Initializes a new instance of BoneWeightCollection.















Public Properties

Name	Description
 Count	(Inherited from Collection .)
 Item	(Inherited from Collection .)







Protected Properties

Name	Description
 Items	(Inherited from Collection .)

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 NormalizeWeights	Overloaded. Normalizes the contents of the bone weights list.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	(Inherited from Collection .)
 Finalize	(Inherited from Object .)
 InsertItem	(Inherited from Collection .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from Collection .)
 SetItem	(Inherited from Collection .)

See Also

Reference

[BoneWeightCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeightCollection Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **BoneWeightCollection**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BoneWeightCollection ()
```

See Also

Reference

[BoneWeightCollection Class](#)
















[BoneWeightCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)







PlatformsWindows XP SP2, Windows Vista

BoneWeightCollection Methods

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	NormalizeWeights	Overloaded. Normalizes the contents of the bone weights list.
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ClearItems	(Inherited from Collection .)
	Finalize	(Inherited from Object .)
	InsertItem	(Inherited from Collection .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	(Inherited from Collection .)
	SetItem	(Inherited from Collection .)

See Also

Reference

[BoneWeightCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeightCollection.NormalizeWeights Method

Normalizes the contents of the bone weights list.

Overload List

Name	Description
BoneWeightCollection.NormalizeWeights ()	Normalizes the contents of the weights list.
BoneWeightCollection.NormalizeWeights (Int32)	Normalizes the contents of the bone weights list.

See Also

Reference

[BoneWeightCollection Class](#)

[BoneWeightCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

BoneWeightCollection.NormalizeWeights Method ()

Note

This method is available only when developing for Windows.

Normalizes the contents of the weights list.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void NormalizeWeights ()
```

Remarks

Normalization does the following:

- Sorts weights such that the most significant weight is first.
- Removes zero-value entries.
- Adjusts values so the sum equals one.

Throws [InvalidContentException](#) if all weights are zero.

See Also

Reference

[BoneWeightCollection Class](#)

[BoneWeightCollection Members](#)

[NormalizeWeights](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoneWeightCollection.NormalizeWeights Method (Int32)

Note

This method is available only when developing for Windows.

Normalizes the contents of the bone weights list.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void NormalizeWeights (  
    int maxWeights  
)
```

Parameters

maxWeights

Maximum number of weights allowed.

Remarks

Normalization does the following:

- Sorts weights such that the most significant weight is first.
- Removes zero-value entries.
- Discards weights with the smallest value until there are *maxWeights* or less in the list.
- Adjusts values so the sum equals one.

Throws [InvalidContentException](#) if all weights are zero.

See Also

Reference

[BoneWeightCollection Class](#)

[BoneWeightCollection Members](#)



[NormalizeWeights](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BoneWeightCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[BoneWeightCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt1BitmapContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing compressed textures (DXT1).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class Dxt1BitmapContent : DxtBitmapContent
```

See Also

Reference

[Dxt1BitmapContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista






Dxt1BitmapContent Members

The following tables list the members exposed by the Dxt1BitmapContent type.







Public Constructors

Name	Description
 Dxt1BitmapContent	Initializes a new instance of Dxt1BitmapContent with the specified width and height.




Public Properties

Name	Description
 Height	(Inherited from BitmapContent .)
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Width	(Inherited from BitmapContent .)

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 TryGetFormat	Attempts to get the GPU texture format of this bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[Dxt1BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt1BitmapContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **Dxt1BitmapContent** with the specified width and height.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Dxt1BitmapContent (  
    int width,  
    int height  
)
```

Parameters

width

Width of the bitmap, in pixels.

height

Height of the bitmap, in pixels.

See Also

Reference

[Dxt1BitmapContent Class](#)







[Dxt1BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)




PlatformsWindows XP SP2, Windows Vista

Dxt1BitmapContent Methods

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 TryGetFormat	Attempts to get the GPU texture format of this bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[Dxt1BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt1BitmapContent.TryGetFormat Method

Note

This method is available only when developing for Windows.

Attempts to get the GPU texture format of this bitmap type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool TryGetFormat (  
    out SurfaceFormat format  
)
```

Parameters

format

[[OutAttribute](#)] Matching GPU format.

Return Value

true if the bitmap matches a GPU format; **false** otherwise.

See Also

Reference

[Dxt1BitmapContent Class](#)






[Dxt1BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Dxt1BitmapContent Properties

Public Properties

	Name	Description
	Height	(Inherited from BitmapContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Width	(Inherited from BitmapContent .)

See Also

Reference

[Dxt1BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt3BitmapContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing compressed textures (DXT3).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class Dxt3BitmapContent : DxtBitmapContent
```

See Also

Reference

[Dxt3BitmapContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista






Dxt3BitmapContent Members

The following tables list the members exposed by the Dxt3BitmapContent type.







Public Constructors

Name	Description
 Dxt3BitmapContent	Initializes a new instance of Dxt3BitmapContent with the specified width and height.




Public Properties

Name	Description
 Height	(Inherited from BitmapContent .)
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Width	(Inherited from BitmapContent .)

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 TryGetFormat	Attempts to get the GPU texture format of this bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[Dxt3BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt3BitmapContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **Dxt3BitmapContent** with the specified width and height.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Dxt3BitmapContent (  
    int width,  
    int height  
)
```

Parameters

width

Width of the bitmap, in pixels.

height

Height of the bitmap, in pixels.

See Also

Reference

[Dxt3BitmapContent Class](#)







[Dxt3BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)




PlatformsWindows XP SP2, Windows Vista

Dxt3BitmapContent Methods

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 TryGetFormat	Attempts to get the GPU texture format of this bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[Dxt3BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt3BitmapContent.TryGetFormat Method

Note

This method is available only when developing for Windows.

Attempts to get the GPU texture format of this bitmap type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool TryGetFormat (  
    out SurfaceFormat format  
)
```

Parameters

format

[[OutAttribute](#)] Matching GPU format.

Return Value

true if the bitmap matches a GPU format; **false** otherwise.

See Also

Reference

[Dxt3BitmapContent Class](#)






[Dxt3BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Dxt3BitmapContent Properties

Public Properties

	Name	Description
	Height	(Inherited from BitmapContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Width	(Inherited from BitmapContent .)

See Also

Reference

[Dxt3BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt5BitmapContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing compressed textures (DXT5).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class Dxt5BitmapContent : DxtBitmapContent
```

See Also

Reference

[Dxt5BitmapContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista






Dxt5BitmapContent Members

The following tables list the members exposed by the Dxt5BitmapContent type.







Public Constructors

Name	Description
 Dxt5BitmapContent	Initializes a new instance of Dxt5BitmapContent with the specified width and height.




Public Properties

Name	Description
 Height	(Inherited from BitmapContent .)
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Width	(Inherited from BitmapContent .)

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 TryGetFormat	Attempts to get the GPU texture format of this bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[Dxt5BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt5BitmapContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **Dxt5BitmapContent** with the specified width and height.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Dxt5BitmapContent (  
    int width,  
    int height  
)
```

Parameters

width

Width of the bitmap, in pixels.

height

Height of the bitmap, in pixels.

See Also

Reference

[Dxt5BitmapContent Class](#)







[Dxt5BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)




PlatformsWindows XP SP2, Windows Vista

Dxt5BitmapContent Methods

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 TryGetFormat	Attempts to get the GPU texture format of this bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[Dxt5BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Dxt5BitmapContent.TryGetFormat Method

Note

This method is available only when developing for Windows.

Attempts to get the GPU texture format of this bitmap type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool TryGetFormat (  
    out SurfaceFormat format  
)
```

Parameters

format

[[OutAttribute](#)] Matching GPU format.

Return Value

true if the bitmap matches a GPU format; **false** otherwise.

See Also

Reference

[Dxt5BitmapContent Class](#)






[Dxt5BitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Dxt5BitmapContent Properties

Public Properties

	Name	Description
	Height	(Inherited from BitmapContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Width	(Inherited from BitmapContent .)

See Also

Reference

[Dxt5BitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

DxtBitmapContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing compressed textures (DXT1, DXT3, DXT5). This is the base class for all compressed textures.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class DxtBitmapContent : BitmapContent
```

See Also

Reference

[DxtBitmapContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista






DxtBitmapContent Members

The following tables list the members exposed by the DxtBitmapContent type.








Public Constructors

Name	Description
 DxtBitmapContent	Overloaded. Initializes a new instance of DxtBitmapContent.






Public Properties

Name	Description
 Height	(Inherited from BitmapContent .)
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Width	(Inherited from BitmapContent .)

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetPixelData	Gets the bitmap content as an array of encoded bytes.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 SetPixelData	Sets the contents of the bitmap using an encoded byte array.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 TryCopyFrom	Attempts to copy from a specified region to another.
 TryCopyTo	Attempts to copy the specified region to another.
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[DxtBitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

DxtBitmapContent Constructor

Initializes a new instance of **DxtBitmapContent**.

Overload List

Name	Description
DxtBitmapContent (Int32)	Initializes a new instance of DxtBitmapContent with the specified compression.
DxtBitmapContent (Int32, Int32, Int32)	Initializes a new instance of DxtBitmapContent with the specified size.

See Also

Reference

[DxtBitmapContent Class](#)

[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

DxtBitmapContent Constructor (Int32)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **DxtBitmapContent** with the specified compression.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected DxtBitmapContent (  
    int blockSize  
)
```

Parameters

blockSize

Size of the block, in bytes.

See Also

Reference

[DxtBitmapContent Class](#)

[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

DxtBitmapContent Constructor (Int32, Int32, Int32)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **DxtBitmapContent** with the specified size.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected DxtBitmapContent (  
    int blockSize,  
    int width,  
    int height  
)
```

Parameters

blockSize

Size of the block, in bytes.

width

Width of the bitmap, in pixels.

height

Height of the bitmap, in pixels.

See Also

Reference

[DxtBitmapContent Class](#)








[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)






PlatformsWindows XP SP2, Windows Vista

DxtBitmapContent Methods

Public Methods

	Name	Description
	Copy	(Inherited from BitmapContent .)
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetPixelData	Gets the bitmap content as an array of encoded bytes.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetPixelData	Sets the contents of the bitmap using an encoded byte array.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	TryCopyFrom	Attempts to copy from a specified region to another.
	TryCopyTo	Attempts to copy the specified region to another.
	ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[DxtBitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

DxtBitmapContent.GetPixelData Method

Note

This method is available only when developing for Windows.

Gets the bitmap content as an array of encoded bytes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override byte[] GetPixelData ()
```

Return Value

Contents of the bitmap.

See Also

Reference

[DxtBitmapContent Class](#)

[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

DxtBitmapContent.SetPixelData Method

Note

This method is available only when developing for Windows.

Sets the contents of the bitmap using an encoded byte array.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override void SetPixelData (  
    byte[] sourceData  
)
```

Parameters

sourceData

Contents to be copied to destination bitmap.

See Also

Reference

[DxtBitmapContent Class](#)

[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

DxtBitmapContent.TryCopyFrom Method

Note

This method is available only when developing for Windows.

Attempts to copy from a specified region to another.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override bool TryCopyFrom (  
    BitmapContent sourceBitmap,  
    Rectangle sourceRegion,  
    Rectangle destinationRegion  
)
```

Parameters

sourceBitmap

Bitmap being copied from.

sourceRegion

Region of bitmap being copied.

destinationRegion

Region being copied to.

Return Value

true if successful; **false** otherwise.

See Also

Reference

[DxtBitmapContent Class](#)

[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

DxtBitmapContent.TryCopyTo Method

Note

This method is available only when developing for Windows.

Attempts to copy the specified region to another.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override bool TryCopyTo (  
    BitmapContent destinationBitmap,  
    Rectangle sourceRegion,  
    Rectangle destinationRegion  
)
```

Parameters

destinationBitmap

Bitmap being copied to.

sourceRegion

Region of source bitmap being copied from.

destinationRegion

Region of destination bitmap.

Return Value

true if successful; **false** otherwise.

See Also

Reference

[DxtBitmapContent Class](#)






[DxtBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

DxtBitmapContent Properties

Public Properties

	Name	Description
	Height	(Inherited from BitmapContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Width	(Inherited from BitmapContent .)

See Also

Reference

[DxtBitmapContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectContent Class

Note

This class is available only when developing for Windows.

Contains the source code for a DirectX Effect, loaded from a .fx file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class EffectContent : ContentItem
```

See Also

Reference

[EffectContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista





EffectContent Members

The following tables list the members exposed by the EffectContent type.






Public Constructors

Name	Description
 EffectContent	Initializes a new instance of EffectContent.



Public Properties

Name	Description
 EffectCode	Gets or sets the effect program source code.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **EffectContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public EffectContent ()
```

See Also

Reference

[EffectContent Class](#)






[EffectContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

EffectContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[EffectContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectContent Properties

Public Properties

	Name	Description
	EffectCode	Gets or sets the effect program source code.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[EffectContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectContent.EffectCode Property

Note

This property is available only when developing for Windows.

Gets or sets the effect program source code.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string EffectCode { get; set; }
```

Property Value

Current value of the Effect program source code or the value to be set.

See Also

Reference

[EffectContent Class](#)

[EffectContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

EffectMaterialContent Class

Note

This class is available only when developing for Windows.

Provides support for representing DirectX [Effect](#) materials.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class EffectMaterialContent : MaterialContent
```

See Also

Reference

[EffectMaterialContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista



EffectMaterialContent Members

The following tables list the members exposed by the EffectMaterialContent type.







Public Constructors

Name	Description
 EffectMaterialContent	Initializes a new instance of EffectMaterialContent.






Public Fields

Name	Description
 CompiledEffectKey	Specifies the key of the key/data pair for the compiled effect property.
 EffectKey	Use this key search for the related effect property in the OpaqueDataDictionary object.








Public Properties

Name	Description
 CompiledEffect	Gets or sets the compiled effect property.
 Effect	Gets or sets the effect property.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Textures	(Inherited from MaterialContent .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetReferenceTypeProperty	(Inherited from MaterialContent .)
 GetTexture	(Inherited from MaterialContent .)
 GetValueTypeProperty	(Inherited from MaterialContent .)
 MemberwiseClone	(Inherited from Object .)
 SetProperty	(Inherited from MaterialContent .)
 SetTexture	(Inherited from MaterialContent .)

See Also



Reference

[EffectMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectMaterialContent Fields

Public Fields

	Name	Description
	CompiledEffectKey	Specifies the key of the key/data pair for the compiled effect property.
	EffectKey	Use this key search for the related effect property in the OpaqueDataDictionary object.

See Also

Reference

[EffectMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectMaterialContent.CompiledEffectKey Field

Note

This field is available only when developing for Windows.

Specifies the key of the key/data pair for the compiled effect property.

This key/data pair is stored in the [OpaqueData](#) property of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string CompiledEffectKey
```

See Also

Reference

[EffectMaterialContent Class](#)

[EffectMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

EffectMaterialContent.EffectKey Field

Note

This field is available only when developing for Windows.

Use this key search for the related effect property in the [OpaqueDataDictionary](#) object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string EffectKey
```

See Also

Reference

[EffectMaterialContent Class](#)

[EffectMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

EffectMaterialContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **EffectMaterialContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public EffectMaterialContent ()
```

See Also

Reference

[EffectMaterialContent Class](#)






[EffectMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)








Platforms Windows XP SP2, Windows Vista

EffectMaterialContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	GetReferenceTypeProperty	(Inherited from MaterialContent .)
	GetTexture	(Inherited from MaterialContent .)
	GetValueTypeProperty	(Inherited from MaterialContent .)
	MemberwiseClone	(Inherited from Object .)
	SetProperty	(Inherited from MaterialContent .)
	SetTexture	(Inherited from MaterialContent .)

See Also







Reference

[EffectMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectMaterialContent Properties

Public Properties

	Name	Description
	CompiledEffect	Gets or sets the compiled effect property.
	Effect	Gets or sets the effect property.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Textures	(Inherited from MaterialContent .)

See Also

Reference

[EffectMaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

EffectMaterialContent.CompiledEffect Property

Note

This property is available only when developing for Windows.

Gets or sets the compiled effect property.

This is a shortcut for accessing **OpaqueData[CompiledEffectKey]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference<CompiledEffect> CompiledEffect { get; set; }
```

Property Value

Reference to a compiled effect or the value to be set.

See Also

Reference

[EffectMaterialContent Class](#)

[EffectMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

EffectMaterialContent.Effect Property

Note

This property is available only when developing for Windows.

Gets or sets the effect property.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ExternalReference<EffectContent> Effect { get; set; }
```

Property Value

Reference to an external DirectX Effect file.

Remarks

Use this value as a shortcut for accessing the [EffectKey](#) element.

See Also

Reference

[EffectMaterialContent Class](#)

[EffectMaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescription Class

Note

This class is available only when developing for Windows.

Provides information to the [FontDescriptionProcessor](#) describing which font to rasterize, which font size to utilize, and which Unicode characters to include in the processor output.

Note

As with most types of software, font files are licensed rather than sold. Font licenses vary from vendor to vendor, but most don't allow redistribution of the fonts, and that includes redistribution of reproductions such as bitmaps containing the rasterized character set. This is even true of many of the licenses covering fonts that Microsoft supplies with applications and Windows. Be careful, therefore, to ensure that you have the required license rights to redistribute any font you include as a bitmap containing the rasterized character set in your game!

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class FontDescription : ContentItem
```

See Also

Reference

[FontDescription Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista











FontDescription Members

The following tables list the members exposed by the FontDescription type.






Public Constructors

Name	Description
 FontDescription	Overloaded. Initializes a new instance of FontDescription.



Public Properties

Name	Description
 Characters	Gets the collection of characters provided by this FontDescription .
 DefaultCharacter	Gets or sets the default character for the font.
 FontName	Gets or sets the name of the font, such as "Times New Roman" or "Arial". This value cannot be null or empty.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Size	Gets or sets the size, in points, of the font.
 Spacing	Gets or sets the amount of space, in pixels, to insert between letters in a string.
 Style	Gets or sets the style of the font, expressed as a combination of one or more FontDescriptionStyle flags.
 UseKerning	Indicates if kerning information is used when drawing characters.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontDescription Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

FontDescription Constructor

Initializes a new instance of **FontDescription**.

Overload List

Name	Description
FontDescription (String, Single, Single)	Initializes a new instance of FontDescription and initializes its members to the specified font, size, and spacing, using FontDescriptionStyle.Regular as the default value for Style .
FontDescription (String, Single, Single, FontDescriptionStyle)	Initializes a new instance of FontDescription and initializes its members to the specified font, size, spacing, and style.
FontDescription (String, Single, Single, FontDescriptionStyle, Boolean)	Initializes a new instance of FontDescription using the specified values.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

FontDescription Constructor (String, Single, Single)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **FontDescription** and initializes its members to the specified font, size, and spacing, using [FontDescriptionStyle.Regular](#) as the default value for *Style*.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontDescription (
    string fontName,
    float size,
    float spacing
)
```

Parameters

fontName

The name of the font, such as *Times New Roman*.

size

The size, in points, of the font.

spacing

The amount of space, in pixels, to insert between letters in a string.

Exceptions

Exception type	Condition
ArgumentNullException	<i>fontName</i> is null or an empty string.
ArgumentOutOfRangeException	<i>size</i> is less than or equal to zero. Specify a value for this parameter that is greater than zero.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescription Constructor (String, Single, Single, FontStyle)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **FontDescription** and initializes its members to the specified font, size, spacing, and style.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontDescription (  
    string fontName,  
    float size,  
    float spacing,  
    FontStyle fontStyle  
)
```

Parameters

fontName

The name of the font, such as *Times New Roman*.

size

The size, in points, of the font.

spacing

The amount of space, in pixels, to insert between letters in a string.

fontStyle

The font style for the font.

Exceptions

Exception type	Condition
ArgumentNullException	<i>fontName</i> is null or an empty string.
ArgumentOutOfRangeException	<i>size</i> is less than or equal to zero. Specify a value for this parameter that is greater than zero.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

FontDescription Constructor (String, Single, Single, FontDescriptionStyle, Boolean)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **FontDescription** using the specified values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontDescription (  
    string fontName,  
    float size,  
    float spacing,  
    FontDescriptionStyle fontStyle,  
    bool useKerning  
)
```

Parameters

fontName

The name of the font, such as *Times New Roman*.

size

The size, in points, of the font.

spacing

The amount of space, in pixels, to insert between letters in a string.

fontStyle

The font style for the font.

useKerning

true if kerning information is used when drawing characters; **false** otherwise.

See Also

Reference

[FontDescription Class](#)






[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

FontDescription Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[FontDescription Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

FontDescription Properties

Public Properties

	Name	Description
	Characters	Gets the collection of characters provided by this FontDescription .
	DefaultCharacter	Gets or sets the default character for the font.
	FontName	Gets or sets the name of the font, such as "Times New Roman" or "Arial". This value cannot be null or empty.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Size	Gets or sets the size, in points, of the font.
	Spacing	Gets or sets the amount of space, in pixels, to insert between letters in a string.
	Style	Gets or sets the style of the font, expressed as a combination of one or more FontDescriptionStyle flags.
	UseKerning	Indicates if kerning information is used when drawing characters.

See Also

Reference

[FontDescription Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

FontDescription.Characters Property

Note

This property is available only when developing for Windows.

Gets the collection of characters provided by this [FontDescription](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentSerializerIgnoreAttribute]  
public ICollection<char> Characters { get; }
```

Property Value

The collection of characters.

Remarks

Characters can be used to change the characters provided by this [FontDescription](#).

The **Characters** collection keeps track of duplicates automatically. It is not necessary for the user to make sure that each letter is added only once. The **Characters** collection will contain only one instance of each character, no matter how many times you call **Add**.

This collection of characters can be used to implement processing for a message file. A custom processor could read the message file, which would contain all of the text that the game will need to print, and add each character it finds to **Characters**.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Concepts

[How To: Extend the Font Description Processor to Support Additional Characters](#)

PlatformsWindows XP SP2, Windows Vista

FontDescription.DefaultCharacter Property

Note

This property is available only when developing for Windows.

Gets or sets the default character for the font.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentSerializerAttribute]  
public Nullable<char> DefaultCharacter { get; set; }
```

Property Value

The default character for this font.

Remarks

If set to anything other than null, this character will be automatically substituted any time an attempt is made to draw or measure characters that are not in the font.

If set to null, missing characters will produce an exception.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

FontDescription.FontName Property

Note

This property is available only when developing for Windows.

Gets or sets the name of the font, such as "Times New Roman" or "Arial". This value cannot be **null** or empty.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string FontName { get; set; }
```

Property Value

The name of the font.

Exceptions

Exception type	Condition
ArgumentNullException	FontName is null or an empty string.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescription.Size Property

Note

This property is available only when developing for Windows.

Gets or sets the size, in points, of the font.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public float Size { get; set; }
```

Property Value

The size, in points, of the font.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	Size is less than or equal to zero. Specify a value for this property that is greater than zero.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

FontDescription.Spacing Property

Note

This property is available only when developing for Windows.

Gets or sets the amount of space, in pixels, to insert between letters in a string.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public float Spacing { get; set; }
```

Property Value

The amount of space, in pixels, to insert between letters in a string.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	Spacing is less than or equal to zero. Specify a value for this property that is greater than zero.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

FontDescription.Style Property

Note

This property is available only when developing for Windows.

Gets or sets the style of the font, expressed as a combination of one or more [FontDescriptionStyle](#) flags.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontDescriptionStyle Style { get; set; }
```

Property Value

The style of the font, expressed as a combination of one or more [FontDescriptionStyle](#) flags.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescription.UseKerning Property

Note

This property is available only when developing for Windows.

Indicates if kerning information is used when drawing characters.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentSerializerAttribute]
public bool UseKerning { get; set; }
```

Property Value

true if kerning information should be used when drawing characters; **false** otherwise.

See Also

Reference

[FontDescription Class](#)

[FontDescription Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescriptionStyle Enumeration

Note

This enumeration is available only when developing for Windows.

Flags that describe style information to be applied to text.

You can combine these flags by using a bitwise OR operator (|).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public enum FontDescriptionStyle
```

Members

Member name	Description
Bold	Bold text.
Italic	Italic text.
Regular	Normal text.

See Also

Reference

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GeometryContent Class

Note

This class is available only when developing for Windows.

Provides properties that define various aspects of a geometry batch.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class GeometryContent : ContentItem
```

Remarks

A geometry batch is a subcomponent of a mesh, representing a single piece of homogeneous geometry that can be submitted to the GPU in a single draw call. It contains an indexed triangle list (using a single material) where all vertices share the same set of data channels. Vertices are made unique if there are differences in any of their data channels. Coordinates that require unique vertices on either side of a join create unique vertices.

See Also

Reference

[GeometryContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista








GeometryContent Members

The following tables list the members exposed by the GeometryContent type.






Public Constructors

Name	Description
 GeometryContent	Creates an instance of GeometryContent .



Public Properties

Name	Description
 Identity	(Inherited from ContentItem .)
 Indices	Gets the list of triangle indices for this geometry batch.
 Material	Gets or sets the material of the parent mesh.
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Parent	Gets or sets the parent MeshContent for this object.
 Vertices	Gets the set of vertex batches for the geometry batch.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GeometryContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

GeometryContent Constructor

Note

This constructor is available only when developing for Windows.

Creates an instance of [GeometryContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public GeometryContent ()
```

See Also

Reference

[GeometryContent Class](#)






[GeometryContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

GeometryContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also








Reference

[GeometryContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

GeometryContent Properties

Public Properties

	Name	Description
	Identity	(Inherited from ContentItem .)
	Indices	Gets the list of triangle indices for this geometry batch.
	Material	Gets or sets the material of the parent mesh.
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Parent	Gets or sets the parent MeshContent for this object.
	Vertices	Gets the set of vertex batches for the geometry batch.

See Also

Reference

[GeometryContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

GeometryContent.Indices Property

Note

This property is available only when developing for Windows.

Gets the list of triangle indices for this geometry batch. Geometry is stored as an indexed triangle list, where each group of three indices defines a single triangle.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IndexCollection Indices { get; }
```

Property Value

Collection of associated indices.

Remarks

Use these indices to reference into the collection of [position indices](#) and [vertex channels](#) of the geometry batch.

See Also

Reference

[GeometryContent Class](#)

[GeometryContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GeometryContent.Material Property

Note

This property is available only when developing for Windows.

Gets or sets the material of the parent mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MaterialContent Material { get; set; }
```

Property Value

Parent mesh of the geometry batch.

See Also

Reference

[GeometryContent Class](#)

[GeometryContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GeometryContent.Parent Property

Note

This property is available only when developing for Windows.

Gets or sets the parent [MeshContent](#) for this object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MeshContent Parent { get; set; }
```

Property Value

Mesh content of the geometry batch.

See Also

Reference

[GeometryContent Class](#)

[GeometryContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GeometryContent.Vertices Property

Note

This property is available only when developing for Windows.

Gets the set of vertex batches for the geometry batch.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexContent Vertices { get; }
```

Property Value

List of associated vertex contents.

See Also

Reference

[GeometryContent Class](#)

[GeometryContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GeometryContentCollection Class

Note

This class is available only when developing for Windows.

Provides methods for maintaining a collection of geometry batches that make up a mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class GeometryContentCollection : ChildCollection<MeshContent, GeometryContent>
```

See Also

Reference

[GeometryContentCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista


GeometryContentCollection Members

The following tables list the members exposed by the GeometryContentCollection type.

















Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)





Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	InsertItem	(Inherited from Collection .)
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	SetItem	(Inherited from Collection .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	GetParent	Gets the parent of a child object.
	MemberwiseClone	(Inherited from Object .)
	SetParent	Sets the parent of the specified child object.

See Also

















Reference

[GeometryContentCollection Class](#)





[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

GeometryContentCollection Methods

Public Methods

Name	Description
 Add	(Inherited from Collection.)
 Clear	(Inherited from Collection.)
 Contains	(Inherited from Collection.)
 CopyTo	(Inherited from Collection.)
 Equals	(Inherited from Object.)
 GetEnumerator	(Inherited from Collection.)
 GetHashCode	(Inherited from Object.)
 GetType	(Inherited from Object.)
 IndexOf	(Inherited from Collection.)
 Insert	(Inherited from Collection.)
 InsertItem	(Inherited from Collection.)
 ReferenceEquals	(Inherited from Object.)
 Remove	(Inherited from Collection.)
 RemoveAt	(Inherited from Collection.)
 SetItem	(Inherited from Collection.)
 ToString	(Inherited from Object.)

Protected Methods

Name	Description
 Finalize	(Inherited from Object.)
 GetParent	Gets the parent of a child object.
 MemberwiseClone	(Inherited from Object.)
 SetParent	Sets the parent of the specified child object.

See Also

Reference

[GeometryContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

GeometryContentCollection.GetParent Method

Note

This method is available only when developing for Windows.

Gets the parent of a child object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override MeshContent GetParent (  
    GeometryContent child  
)
```

Parameters

child

The child of the parent being retrieved.

Return Value

The parent of the child object.

See Also

Reference

[GeometryContentCollection Class](#)

[GeometryContentCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

GeometryContentCollection.SetParent Method

Note

This method is available only when developing for Windows.

Sets the parent of the specified child object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void SetParent (  
    GeometryContent child,  
    MeshContent parent  
)
```

Parameters

child

The child of the parent being retrieved.

parent

The parent of the child object.

See Also

Reference

[GeometryContentCollection Class](#)



[GeometryContentCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GeometryContentCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[GeometryContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndexCollection Class

Note

This class is available only when developing for Windows.

Provides methods for maintaining a list of index values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class IndexCollection : Collection<int>
```

See Also

Reference

[IndexCollection Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



IndexCollection Members

The following tables list the members exposed by the IndexCollection type.


Public Constructors

Name	Description
 IndexCollection	Initializes a new instance of IndexCollection.
















Public Properties

Name	Description
 Count	(Inherited from Collection .)
 Item	(Inherited from Collection .)







Protected Properties

Name	Description
 Items	(Inherited from Collection .)

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 AddRange	Appends the specified indices into the collection.
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	(Inherited from Collection .)
 Finalize	(Inherited from Object .)
 InsertItem	(Inherited from Collection .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from Collection .)
 SetItem	(Inherited from Collection .)

See Also

Reference

[IndexCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndexCollection Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **IndexCollection**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IndexCollection ()
```

See Also

Reference

[IndexCollection Class](#)
















[IndexCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)







PlatformsWindows XP SP2, Windows Vista

IndexCollection Methods

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	AddRange	Appends the specified indices into the collection.
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ClearItems	(Inherited from Collection .)
	Finalize	(Inherited from Object .)
	InsertItem	(Inherited from Collection .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	(Inherited from Collection .)
	SetItem	(Inherited from Collection .)

See Also

Reference

[IndexCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndexCollection.AddRange Method

Note

This method is available only when developing for Windows.

Appends the specified indices into the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void AddRange (  
    IEnumerable<int> indices  
)
```

Parameters

indices

Index into the [Positions](#) member of the parent.

See Also

Reference

[IndexCollection Class](#)



[IndexCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndexCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[IndexCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndirectPositionCollection Class

Note

This class is available only when developing for Windows.

Provides methods for maintaining a list of vertex positions.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class IndirectPositionCollection : IList<Vector3>, ICollection<Vector3>, IEnumerable<Vector3>, IEnumerable
```

Remarks

This class is designed to collect the vertex positions for a [VertexContent](#) object. Use the contents of the [PositionIndices](#) property (of the contained [VertexContent](#) object) to index into the [Positions](#) property of the parent mesh.

See Also

Reference

[IndirectPositionCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista










IndirectPositionCollection Members

The following tables list the members exposed by the IndirectPositionCollection type.



Public Properties

Name	Description
 Count	Number of positions in the collection.
 Item	Gets or sets the position at the specified index.








Public Methods

Name	Description
 Contains	Determines whether the specified position is in the collection.
 CopyTo	Copies the specified positions to an array, starting at the specified index.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator interface for reading the position values.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Gets the index of the specified position in a collection.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.IsReadOnly	Gets a value indicating whether this object is read-only.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Add	Adds a new element to the end of the collection.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Clear	Removes all elements from the collection.
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that can iterate through the collection.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Insert	Inserts a new element into the collection.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.RemoveAt	Removes the element at the specified index position.
 System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Remove	Removes the specified element from the collection.

See Also










Reference

[IndirectPositionCollection Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndirectPositionCollection Methods








Public Methods

Name	Description
 Contains	Determines whether the specified position is in the collection.
 CopyTo	Copies the specified positions to an array, starting at the specified index.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator interface for reading the position values.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Gets the index of the specified position in a collection.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic ICollection<Microsoft.Xna.Framework.Vector3>.IsReadOnly	Gets a value indicating whether this object is read-only.
 System.Collections.Generic ICollection<Microsoft.Xna.Framework.Vector3>.Add	Adds a new element to the end of the collection.
 System.Collections.Generic ICollection<Microsoft.Xna.Framework.Vector3>.Clear	Removes all elements from the collection.
 System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that can iterate through the collection.
 System.Collections.Generic IList<Microsoft.Xna.Framework.Vector3>.Insert	Inserts a new element into the collection.
 System.Collections.Generic IList<Microsoft.Xna.Framework.Vector3>.RemoveAt	Removes the element at the specified index position.
 System.Collections.Generic ICollection<Microsoft.Xna.Framework.Vector3>.Remove	Removes the specified element from the collection.

See Also

Reference

[IndirectPositionCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndirectPositionCollection.Contains Method

Note

This method is available only when developing for Windows.

Determines whether the specified position is in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Contains (  
    Vector3 item  
)
```

Parameters

item

Position being searched for in the collection.

Return Value

true if the position was found; **false** otherwise.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndirectPositionCollection.CopyTo Method

Note

This method is available only when developing for Windows.

Copies the specified positions to an array, starting at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void CopyTo (  
    Vector3[] array,  
    int arrayIndex  
)
```

Parameters

array

Array of positions to be copied.

arrayIndex

Index of the first copied position.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

IndirectPositionCollection.GetEnumerator Method

Note

This method is available only when developing for Windows.

Gets an enumerator interface for reading the position values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerator<Vector3> GetEnumerator ()
```

Return Value

Interface for enumerating the collection of position values.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndirectPositionCollection.IndexOf Method

Note

This method is available only when developing for Windows.

Gets the index of the specified position in a collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int IndexOf (  
    Vector3 item  
)
```

Parameters

item

Position being searched for.

Return Value

Index of the specified position.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Add Method

Note

This method is available only when developing for Windows.

Adds a new element to the end of the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Add (  
    Vector3 item  
)
```

Parameters

item

The item to add.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Clear Method

Note

This method is available only when developing for Windows.

Removes all elements from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Clear ()
```

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Remove Method

Note

This method is available only when developing for Windows.

Removes the specified element from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.Remove (  
    Vector3 item  
)
```

Parameters

item

The item to be removed.

Return Value

true if the element was removed; **false** otherwise.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.IList<Microsoft.Xna.Framework.Vector3>.Insert Method

Note

This method is available only when developing for Windows.

Inserts a new element into the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<Microsoft.Xna.Framework.Vector3>.Insert (  
    int index,  
    Vector3 item  
)
```

Parameters

index

Index for element insertion.

item

Element to insert.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.IList<Microsoft.Xna.Framework.Vector3>.RemoveAt Method

Note

This method is available only when developing for Windows.

Removes the element at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<Microsoft.Xna.Framework.Vector3>.RemoveAt (
    int index
)
```

Parameters

index

Index of the element to be removed.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.IEnumerable.GetEnumerator Method

Note

This method is available only when developing for Windows.

Returns an enumerator that can iterate through the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

Enumerator that can iterate through the collection.

See Also

Reference

[IndirectPositionCollection Class](#)



[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndirectPositionCollection Properties

Public Properties

	Name	Description
	Count	Number of positions in the collection.
	Item	Gets or sets the position at the specified index.

See Also

Reference

[IndirectPositionCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

IndirectPositionCollection.Count Property

Note

This property is available only when developing for Windows.

Number of positions in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of positions.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndirectPositionCollection.Item Property

Note

This property is available only when developing for Windows.

Gets or sets the position at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Vector3 this [
    int index
] { get; set; }
```

Property Value

Position located at *index*.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IndirectPositionCollection.System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3> Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether this object is read-only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Vector3>.IsReadOnly { get; }
```

Property Value

true if this object is read-only; **false** otherwise.

See Also

Reference

[IndirectPositionCollection Class](#)

[IndirectPositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for maintaining a collection of named texture references.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class MaterialContent : ContentItem
```

Remarks

In addition to texture references, opaque data values are stored in the [OpaqueData](#) property of the base class.

See Also

Reference

[MaterialContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista





MaterialContent Members

The following tables list the members exposed by the MaterialContent type.






Public Constructors

Name	Description
 MaterialContent	Initializes a new instance of MaterialContent .








Public Properties

Name	Description
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Textures	Gets the texture collection of the material.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetReferenceTypeProperty	Gets a reference type from the OpaqueDataDictionary collection.
 GetTexture	Gets a value from the Textures collection.
 GetValueTypeProperty	Gets a value type from the OpaqueDataDictionary collection.
 MemberwiseClone	(Inherited from Object .)
 SetProperty	Sets a value in the contained OpaqueDataDictionary object.
 SetTexture	Sets a value in the contained TextureReferenceDictionary object.

See Also

Reference

[MaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MaterialContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [MaterialContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MaterialContent ()
```

See Also

Reference

[MaterialContent Class](#)






[MaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)








PlatformsWindows XP SP2, Windows Vista

MaterialContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	GetReferenceTypeProperty	Gets a reference type from the OpaqueDataDictionary collection.
	GetTexture	Gets a value from the Textures collection.
	GetValueTypeProperty	Gets a value type from the OpaqueDataDictionary collection.
	MemberwiseClone	(Inherited from Object .)
	 SetProperty	Sets a value in the contained OpaqueDataDictionary object.
	SetTexture	Sets a value in the contained TextureReferenceDictionary object.

See Also

Reference

[MaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MaterialContent.GetReferenceTypeProperty Generic Method

Note

This generic method is available only when developing for Windows.

Gets a reference type from the [OpaqueDataDictionary](#) collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected T GetReferenceTypeProperty<T> (  
    string key  
)
```

Type Parameters

T

Parameters

key

Key of the property being retrieved.

Return Value

Type of the related opaque data.

See Also

Reference

[MaterialContent Class](#)

[MaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialContent.GetTexture Method

Note

This method is available only when developing for Windows.

Gets a value from the [Textures](#) collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ExternalReference<TextureContent> GetTexture (  
    string key  
)
```

Parameters

key

Key of the texture being retrieved.

Return Value

Reference to a texture from the collection.

See Also

Reference

[MaterialContent Class](#)

[MaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialContent.GetValueTypeProperty Generic Method

Note

This generic method is available only when developing for Windows.

Gets a value type from the [OpaqueDataDictionary](#) collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected Nullable<T> GetValueTypeProperty<T> (  
    string key  
) where T : ValueType
```

Type Parameters

T

Parameters

key

Key of the value type being retrieved.

Return Value

Index of the value type being retrieved.

See Also

Reference

[MaterialContent Class](#)

[MaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialContent.SetProperty Generic Method

Note

This generic method is available only when developing for Windows.

Sets a value in the contained [OpaqueDataDictionary](#) object.

If **null** is passed, the value is removed.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected void SetProperty<T> (  
    string key,  
    T value  
)
```

Type Parameters

T

Type of the element being set.

Parameters

key

Name of the key being modified.

value

Value being set.

See Also

Reference

[MaterialContent Class](#)

[MaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialContent.SetTexture Method

Note

This method is available only when developing for Windows.

Sets a value in the contained [TextureReferenceDictionary](#) object.

If **null** is passed, the value is removed.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected void SetTexture (  
    string key,  
    ExternalReference<TextureContent> value  
)
```

Parameters

key

Name of the key being modified.

value

Value being set.

Remarks

The *key* value differs depending on the type of attached dictionary.

If attached to a [BasicMaterialContent](#) dictionary (which becomes a [BasicEffect](#) object at run time), the value for the Texture key is used as the texture for the [BasicEffect](#) runtime object. Other keys are ignored.

If attached to a [EffectMaterialContent](#) dictionary, key names are the texture names used by the effect. These names are dependent upon the author of the effect object.

See Also

Reference

[MaterialContent Class](#)

[MaterialContent Members](#)





[Texture](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MaterialContent Properties

Public Properties

	Name	Description
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Textures	Gets the texture collection of the material.

See Also

Reference

[MaterialContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MaterialContent.Textures Property

Note

This property is available only when developing for Windows.

Gets the texture collection of the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TextureReferenceDictionary Textures { get; }
```

Property Value

Collection of textures used by the material.

See Also

Reference

[MaterialContent Class](#)

[MaterialContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshBuilder Class

Note

This class is available only when developing for Windows.

Provides support for writing a custom importer for mesh objects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class MeshBuilder
```

Remarks

MeshBuilder is designed to make it easier to write importers. It creates an abstraction of the internal workings of the [MeshContent](#) and [GeometryContent](#) classes and provides a simple way to create a mesh object.

There are three steps to creating a mesh.

1. Call [StartMesh](#) to get a [MeshBuilder](#) object. To initialize this object, fill the positions buffer with data by using the [CreatePosition](#) function. After retrieving the positions data, call the [CreateVertexChannel](#) generic function to specify the types of vertex channels needed—for example, normals, UVs, and color channels.

Note

The [PositionCollection](#) property of a [MeshContent](#) object is created automatically. Therefore, there is no need to explicitly declare a [VertexChannel](#) object for positions.

2. After setting up the position and vertex data channel buffers, begin creating triangles. Use [SetMaterial](#) and [SetOpaqueData](#) to set the data of each triangle, and use [SetVertexChannelData](#) to set the individual vertex data of each triangle. After setting this data, call [AddTriangleVertex](#) for each vertex of each triangle.

Note

If you call [CreatePosition](#) or [CreateVertexChannel](#) after calling [AddTriangleVertex](#), an [InvalidOperationException](#) exception is thrown.

3. In the final step, call [FinishMesh](#). After this call, the mesh is optimized automatically with calls to [MergeDuplicateVertices](#) and [CalculateNormals](#).

For a complete example of this procedure, download the [Generated Geometry Sample](#) from the Creators Club Online website and load the solution. Open the TerrainProcessor.cs file and examine the `Process` method. This method implements the steps listed above.

See Also

Conceptual

[How To: Write a Custom Importer and Processor](#)

Reference

[MeshBuilder Members](#)

[GeometryContent](#)





[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista















MeshBuilder Members

The following tables list the members exposed by the MeshBuilder type.



Public Properties

	Name	Description
	MergeDuplicatePositions	Gets or sets the current value for position merging of the mesh.
	MergePositionTolerance	Gets or sets the tolerance for MergeDuplicatePositions .
	Name	Gets or sets the name of the current MeshContent object being processed.
	SwapWindingOrder	Reverses the triangle winding order of the specified mesh.

Public Methods

	Name	Description
	AddTriangleVertex	Adds a vertex into the index collection.
	CreatePosition	Overloaded. Inserts the specified vertex position into the vertex channel.
	CreateVertexChannel	Creates a vertex data channel for use by the mesh.
	Equals	(Inherited from Object .)
	FinishMesh	Ends the creation of a mesh.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetMaterial	Specifies the material used by the current mesh.
	SetOpaqueData	Initializes the opaque data for a specific mesh object.
	SetVertexChannelData	Sets the specified vertex data with new data.
 	StartMesh	Initializes the creation of a mesh.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also















Reference

[MeshBuilder Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshBuilder Methods

Public Methods

	Name	Description
	AddTriangleVertex	Adds a vertex into the index collection.
	CreatePosition	Overloaded. Inserts the specified vertex position into the vertex channel.
	CreateVertexChannel	Creates a vertex data channel for use by the mesh.
	Equals	(Inherited from Object .)
	FinishMesh	Ends the creation of a mesh.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	SetMaterial	Specifies the material used by the current mesh.
	SetOpaqueData	Initializes the opaque data for a specific mesh object.
	SetVertexChannelData	Sets the specified vertex data with new data.
 	StartMesh	Initializes the creation of a mesh.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MeshBuilder Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshBuilder.AddTriangleVertex Method

Note

This method is available only when developing for Windows.

Adds a vertex into the index collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void AddTriangleVertex (  
    int indexIntoVertexCollection  
)
```

Parameters

indexIntoVertexCollection

Index of the inserted vertex, in the collection. This corresponds to the value returned by [CreatePosition](#).

Remarks

[MeshBuilder](#) supports triangle lists only. Therefore, calls to **AddTriangleVertex** must occur in groups of three. In addition, [MeshBuilder](#) automatically determines which [GeometryContent](#) object receives the current triangle based on the state data. This data is set by the last calls to [SetMaterial](#) and [SetOpaqueData](#).

This means that per-vertex material and opaque data are not supported. Therefore, do not call [SetMaterial](#) or [SetOpaqueData](#) when defining a triangle. For an example of the proper method, see the [MeshBuilder](#) class overview.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshBuilder.CreatePosition Method

Inserts the specified vertex position into the vertex channel.

Overload List

Name	Description
MeshBuilder.CreatePosition (Single, Single, Single)	Inserts the specified vertex position into the vertex channel.
MeshBuilder.CreatePosition (Vector3)	Inserts the specified vertex position into the vertex channel at the specified index.

Remarks

These vertices are shared between the mesh's geometry objects, preserving the topology information. For example, two different geometries could reference the same position, but have a different material, texture coordinates, and normals.

In most cases, the index returned by **CreatePosition** is based on the number of times this function has been called. For example, **CreatePosition** returns 0 on the first call, 1 on the second call, and so forth. Therefore, if you call **CreatePosition** for each vertex in your vertex collection, your vertex collection and the [Positions](#) collection of the finished [MeshContent](#) object are parallel, assuming [MergeDuplicatePositions](#) doesn't find anything to merge.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshBuilder.CreatePosition Method (Single, Single, Single)

Note

This method is available only when developing for Windows.

Inserts the specified vertex position into the vertex channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int CreatePosition (  
    float x,  
    float y,  
    float z  
)
```

Parameters

x

Value of the x component of the vector.

y

Value of the y component of the vector.

z

Value of the z component of the vector.

Return Value

Index of the inserted vertex.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder.CreatePosition Method (Vector3)

Note

This method is available only when developing for Windows.

Inserts the specified vertex position into the vertex channel at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int CreatePosition (  
    Vector3 pos  
)
```

Parameters

pos

Value of the vertex being inserted.

Return Value

Index of the vertex being inserted.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder.CreateVertexChannel Generic Method

Note

This generic method is available only when developing for Windows.

Creates a vertex data channel for use by the mesh. Typically, the data channel holds texture coordinates, normals, and other per-vertex data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int CreateVertexChannel<T> (  
    string usage  
)
```

Type Parameters

T

Type of the data channel.

Parameters

usage

Describes how the channel is used. Use this parameter as a look-up into a [VertexChannelCollection](#). It is strongly recommended that you choose a *usage* from the following list.

- Binormal
- Color
- Normal
- Tangent
- TextureCoordinate
- Weights

For more information, see [VertexChannelNames](#).

Return Value

Index identifying the new data channel. Use this as a parameter when calling [SetVertexChannelData](#).

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder.FinishMesh Method

Note

This method is available only when developing for Windows.

Ends the creation of a mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MeshContent FinishMesh ()
```

Return Value

Resultant mesh.

Remarks

Once called, **FinishMesh** optimizes the related mesh by calling [MergeDuplicateVertices](#). In addition, mesh normals are generated (by calling [CalculateNormals](#)) if none currently exist.

Note

The related [MeshBuilder](#) object can no longer be used after this function is called.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[StartMesh](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder.SetMaterial Method

Note

This method is available only when developing for Windows.

Specifies the material used by the current mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void SetMaterial (  
    MaterialContent material  
)
```

Parameters

material

The material to be used by the mesh.

Remarks

Sets the material used by the triangle being defined next. This material, in conjunction with [SetOpaqueData](#), determines the [GeometryContent](#) object containing the next triangle. [MeshBuilder](#) maintains the material value for all future triangles. Therefore, if multiple triangles share the same material, you need only one call to [SetMaterial](#).

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder.SetOpaqueData Method

Note

This method is available only when developing for Windows.

Initializes the opaque data for a specific mesh object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void SetOpaqueData (  
    OpaqueDataDictionary opaqueData  
)
```

Parameters

opaqueData

Opaque data to be applied to the [GeometryContent](#) object of the next triangle.

Remarks

The opaque data is used in conjunction with the current material (set with a call to [SetMaterial](#)) to determine the [GeometryContent](#) object that will contain the next triangle. [MeshBuilder](#) maintains the material value for all future triangles. Therefore, if multiple triangles share the same material, you need only one call to [SetMaterial](#).

SetOpaqueData can be called only on [GeometryContent](#) and [MeshContent](#) objects.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshBuilder.SetVertexChannelData Method

Note

This method is available only when developing for Windows.

Sets the specified vertex data with new data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void SetVertexChannelData (
    int vertexDataIndex,
    Object channelData
)
```

Parameters

vertexDataIndex

Index of the vertex data channel being set. This should match the index returned by [CreateVertexChannel](#).

channelData

New data values for the vertex data. The data type being set must match the data type for the vertex channel specified by *vertexDataIndex*.

Remarks

The following table lists the exceptions that can be thrown by this method:

Exception	Cause
KeyNotFoundException	If <i>vertexDataIndex</i> does not correspond to an actual vertex data channel created using CreateVertexChannel .
InvalidOperationException	If the type of <i>channelData</i> does not match the data type for the vertex channel specified by <i>vertexDataIndex</i> .

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshBuilder.StartMesh Method

Note

This method is available only when developing for Windows.

Initializes the creation of a mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static MeshBuilder StartMesh (  
    string name  
)
```

Parameters

name

Name of the mesh.

Return Value

Object used when building the mesh.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)





[FinishMesh](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder Properties

Public Properties

	Name	Description
	MergeDuplicatePositions	Gets or sets the current value for position merging of the mesh.
	MergePositionTolerance	Gets or sets the tolerance for MergeDuplicatePositions .
	Name	Gets or sets the name of the current MeshContent object being processed.
	SwapWindingOrder	Reverses the triangle winding order of the specified mesh.

See Also

Reference

[MeshBuilder Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshBuilder.MergeDuplicatePositions Property

Note

This property is available only when developing for Windows.

Gets or sets the current value for position merging of the mesh.

This can be used to preserve topology of unique vertices.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool MergeDuplicatePositions { get; set; }
```

Property Value

Current value of the property.

Remarks

If this property is enabled, every call to [MeshBuilder.CreatePosition \(Vector3\)](#) will check to see whether a position exists for which each component of that position varies no more than the current value of [MergePositionTolerance](#) from the new position. If one is found, the original position is used.

This property should be used in conjunction with [MergePositionTolerance](#). The following code demonstrates this:

C#

```
MeshBuilder b = MeshBuilder.StartMesh( "bill" );  
b.MergeDuplicatePositions = true;  
b.MergePositionTolerance = 1.0f;  
  
// Additional mesh building code here.  
  
MeshContent mesh = b.FinishMesh();
```

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshBuilder.MergePositionTolerance Property

Note

This property is available only when developing for Windows.

Gets or sets the tolerance for [MergeDuplicatePositions](#). The default value is 0.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public float MergePositionTolerance { get; set; }
```

Property Value

Tolerance value used by the mesh.

Remarks

When [MergeDuplicatePositions](#) is enabled, each call to [MeshBuilder.CreatePosition \(Vector3\)](#) checks to see whether a position exists for which each component of that position varies no more than the current value of **MergePositionTolerance** from the new position. If one is found, the original position will be reused.

Note

Setting **MergePositionTolerance** does not also enable position merging. This must be enabled explicitly by calling [MergeDuplicatePositions](#).

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshBuilder.Name Property

Note

This property is available only when developing for Windows.

Gets or sets the name of the current [MeshContent](#) object being processed. This is the same name as was passed in to [StartMesh](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string Name { get; set; }
```

Property Value

Name of the [MeshContent](#) object.

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshBuilder.SwapWindingOrder Property

Note

This property is available only when developing for Windows.

Reverses the triangle winding order of the specified mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool SwapWindingOrder { get; set; }
```

Property Value

Current value of the winding order.

Remarks

If enabled, the triangles of the finished mesh are in the opposite order that they were given to the mesh builder. For example, if the vertices of two triangles were given as 123 456, they would be changed to 321 654. This can be useful if the source data was constructed with back-face culling set to something other than what the destination game engine expects.

The winding order is swapped when [FinishMesh](#) is called. Therefore, it is not necessary to modify the indices returned by [CreatePosition](#).

See Also

Reference

[MeshBuilder Class](#)

[MeshBuilder Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshContent Class

Note

This class is available only when developing for Windows.

Provides properties and methods that define various aspects of a mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class MeshContent : NodeContent
```

Remarks

A mesh has the following characteristics:

- It is not jointed. For example, the wheels of a car are separate meshes from the body mesh of the same car.
- It may contain multiple materials. For example, the car body can be a single mesh even if the windshield component uses a different texture and shader as compared to the leather seat component of the car body.
- Vertices containing different channels may be mixed in a single mesh. For example, the seat geometry includes tangent vectors but the windshield does not. However, using a single mesh preserves topological continuity (shared vertex identities) across material and vertex format boundaries.

See Also

Reference

[MeshContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista











MeshContent Members

The following tables list the members exposed by the MeshContent type.






Public Constructors

Name	Description
 MeshContent	Initializes a new instance of MeshContent .



Public Properties

Name	Description
 AbsoluteTransform	(Inherited from NodeContent .)
 Animations	(Inherited from NodeContent .)
 Children	(Inherited from NodeContent .)
 Geometry	Gets the list of geometry batches for the mesh.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Parent	(Inherited from NodeContent .)
 Positions	Gets the list of vertex position values.
 Transform	(Inherited from NodeContent .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MeshContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of [MeshContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MeshContent ()
```

See Also

Reference

[MeshContent Class](#)






[MeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

MeshContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[MeshContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshContent Properties

Public Properties

	Name	Description
	AbsoluteTransform	(Inherited from NodeContent .)
	Animations	(Inherited from NodeContent .)
	Children	(Inherited from NodeContent .)
	Geometry	Gets the list of geometry batches for the mesh.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Parent	(Inherited from NodeContent .)
	Positions	Gets the list of vertex position values.
	Transform	(Inherited from NodeContent .)

See Also

Reference

[MeshContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshContent.Geometry Property

Note

This property is available only when developing for Windows.

Gets the list of [geometry batches](#) for the mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public GeometryContentCollection Geometry { get; }
```

Property Value

Collection of geometry batches.

See Also

Reference

[MeshContent Class](#)

[MeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshContent.Positions Property

Note

This property is available only when developing for Windows.

Gets the list of vertex position values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PositionCollection Positions { get; }
```

Property Value

Collection of vertex positions.

See Also

Reference

[MeshContent Class](#)

[MeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper Class

Note

This class is available only when developing for Windows.

Provides methods for manipulating mesh data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static class MeshHelper
```

See Also

Reference

[MeshHelper Members](#)















[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



MeshHelper Members

The following tables list the members exposed by the MeshHelper type.

Public Methods

	Name	Description
	CalculateNormals	Computes new normals for the specified mesh.
	CalculateTangentFrames	Compute tangent frames for the given mesh.
	Equals	(Inherited from Object .)
	FindSkeleton	Searches for the root bone of the contained skeleton.
	FlattenSkeleton	Gets a flattened list of all bones contained by the specified skeleton.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MergeDuplicatePositions	Processes the PositionCollection collection of the specified mesh, merging any values that are closer together than the specified tolerance threshold.
	MergeDuplicateVertices	Overloaded. Processes the Indices and VertexChannel data of the specified mesh or geometry batch, merging any duplicate vertices.
	OptimizeForCache	Reorders the indices and vertices of the mesh for optimal GPU cache performance.
	ReferenceEquals	(Inherited from Object .)
	SwapWindingOrder	Reverses the triangle winding order of the specified mesh.
	ToString	(Inherited from Object .)
	TransformScene	Applies a transformation to the contents of a scene hierarchy.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also















Reference

[MeshHelper Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshHelper Methods

Public Methods

	Name	Description
	CalculateNormals	Computes new normals for the specified mesh.
	CalculateTangentFrames	Compute tangent frames for the given mesh.
	Equals	(Inherited from Object .)
	FindSkeleton	Searches for the root bone of the contained skeleton.
	FlattenSkeleton	Gets a flattened list of all bones contained by the specified skeleton.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MergeDuplicatePositions	Processes the PositionCollection collection of the specified mesh, merging any values that are closer together than the specified tolerance threshold.
	MergeDuplicateVertices	Overloaded. Processes the Indices and VertexChannel data of the specified mesh or geometry batch, merging any duplicate vertices.
	OptimizeForCache	Reorders the indices and vertices of the mesh for optimal GPU cache performance.
	ReferenceEquals	(Inherited from Object .)
	SwapWindingOrder	Reverses the triangle winding order of the specified mesh.
	ToString	(Inherited from Object .)
	TransformScene	Applies a transformation to the contents of a scene hierarchy.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MeshHelper Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshHelper.CalculateNormals Method

Note

This method is available only when developing for Windows.

Computes new normals for the specified mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void CalculateNormals (  
    MeshContent mesh,  
    bool overwriteExistingNormals  
)
```

Parameters

mesh

Mesh containing the normals being computed.

overwriteExistingNormals

true if existing normals are to be discarded; **false** otherwise. Existing normals are searched for as if they are a vertex data channel (of type [Vector3 Structure](#)) with the name [VertexChannelNames.Normal](#).

Remarks

The new normals are stored in vertex data channels (of type [Vector3 Structure](#)) with the name [VertexChannelNames.Normal](#).

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshHelper.CalculateTangentFrames Method

Note

This method is available only when developing for Windows.

Compute tangent frames for the given mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void CalculateTangentFrames (
    MeshContent mesh,
    string textureCoordinateChannelName,
    string tangentChannelName,
    string binormalChannelName
)
```

Parameters

mesh

The target mesh used to create the tangent frame. All geometries in this mesh must have normal vertex channels stored in [Normal](#), and must contain [Vector3](#) data.

textureCoordinateChannelName

The texture coordinate channel used for computing the tangent frame. This channel must contain [Vector2](#) data.

tangentChannelName

Target channel name used to store calculated tangents. A tangent channel is not generated if **null** or an empty string is specified.

binormalChannelName

Target channel name used to store calculated binormals. A tangent channel is not generated if **null** or an empty string is specified.

Remarks

This method computes and adds tangent and binormal vertex channels to the given mesh.

An [InvalidContentException](#) is thrown if:

- The mesh does not contain a normal channel (stored in [Normal](#)).
- The data specified in the normal channel is not a [Vector3](#) type.
- The vertex channel of the mesh does not have the name specified by *textureCoordinateChannelName*.
- The data specified in the texture coordinate channel is not a [Vector2](#) type.
- The channel specified by *tangentChannelName* already exists.
- The channel specified by *binormalChannelName* already exists.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.FindSkeleton Method

Note

This method is available only when developing for Windows.

Searches for the root bone of the contained skeleton. This node is used to animate the skeleton.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static BoneContent FindSkeleton (  
    NodeContent node  
)
```

Parameters

node

Mesh that possibly contains the skeleton.

Return Value

Root of the contained skeleton; otherwise **null** if no skeleton is found.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.FlattenSkeleton Method

Note

This method is available only when developing for Windows.

Gets a flattened list of all bones contained by the specified skeleton.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static IList<BoneContent> FlattenSkeleton (  
    BoneContent skeleton  
)
```

Parameters

skeleton

Skeleton to be traversed.

Return Value

List of bones for the animation skeleton.

Remarks

The tree is calculated using a depth-first traversal. The bones are returned in the same order used by [ModelProcessor](#) to convert [BoneWeight](#) vertex channels into the [BlendIndices](#) format.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.MergeDuplicatePositions Method

Note

This method is available only when developing for Windows.

Processes the [PositionCollection](#) collection of the specified mesh, merging any values that are closer together than the specified tolerance threshold.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void MergeDuplicatePositions (  
    MeshContent mesh,  
    float tolerance  
)
```

Parameters

mesh

Mesh to be processed.

tolerance

Tolerance to be used when determining duplicate positions. Pass a value of 0 to merge only those values that are exactly identical.

Remarks

This method also updates the [PositionIndices](#) data to match the modified position table. This should generally be followed by a call to [MergeDuplicateVertices](#).

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.MergeDuplicateVertices Method

Processes the [Indices](#) and [VertexChannel](#) data of the specified mesh or geometry batch, merging any duplicate vertices.

Overload List

Name	Description
MeshHelper.MergeDuplicateVertices (GeometryContent)	Processes the PositionIndices and VertexChannel data of the specified geometry batch, merging any duplicate vertices.
MeshHelper.MergeDuplicateVertices (MeshContent)	Processes the PositionIndices and VertexChannel data of the specified mesh, merging any duplicate vertices.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MeshHelper.MergeDuplicateVertices Method (GeometryContent)

Note

This method is available only when developing for Windows.

Processes the [PositionIndices](#) and [VertexChannel](#) data of the specified geometry batch, merging any duplicate vertices.

This should generally be followed by a call to [MergeDuplicatePositions](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void MergeDuplicateVertices (  
    GeometryContent geometry  
)
```

Parameters

geometry

Geometry batch to be processed.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.MergeDuplicateVertices Method (MeshContent)

Note

This method is available only when developing for Windows.

Processes the [PositionIndices](#) and [VertexChannel](#) data of the specified mesh, merging any duplicate vertices.

This should generally be followed by a call to [MergeDuplicatePositions](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void MergeDuplicateVertices (  
    MeshContent mesh  
)
```

Parameters

mesh

Mesh to be processed.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.OptimizeForCache Method

Note

This method is available only when developing for Windows.

Reorders the indices and vertices of the mesh for optimal GPU cache performance.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void OptimizeForCache (  
    MeshContent mesh  
)
```

Parameters

mesh

Mesh to be optimized.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MeshHelper.SwapWindingOrder Method

Note

This method is available only when developing for Windows.

Reverses the triangle winding order of the specified mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void SwapWindingOrder (  
    MeshContent mesh  
)
```

Parameters

mesh

The mesh to be modified.

Remarks

Call this method when moving data between right-handed and left-handed coordinate systems, or when changing the direction of the back-face culling.

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MeshHelper.TransformScene Method

Note

This method is available only when developing for Windows.

Applies a transformation to the contents of a scene hierarchy.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void TransformScene (  
    NodeContent scene,  
    Matrix transform  
)
```

Parameters

scene

Scene hierarchy being transformed.

transform

Matrix used in the transformation

Remarks

The resulting world space positions are similar to the results obtained from applying the specified transform to the [Transform](#) property of the scene object. However, this method performs the transformation by cascading down through the scene hierarchy and modifying the actual underlying vertex positions. This does not introduce unwanted artifacts into the node transforms. For instance, this method can be used to mirror a scene without leaving mirroring in the transform matrices.

Typically, you would convert a scene from a left-handed to a right-handed coordinate system and then transform it with the following matrix ((1, 0, 0), (0, 1, 0), (0, 0, -1)).

See Also

Reference

[MeshHelper Class](#)

[MeshHelper Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChain Class

Note

This class is available only when developing for Windows.

Provides methods for accessing a mipmap chain.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class MipmapChain : Collection<BitmapContent>
```

Remarks

A mipmap chain consists of one or more individual 2D bitmaps, holding scaled versions of the same underlying image. For example, if the first entry was sized 256×128, the second would be 128×64, and the third 64×32.

See Also

Reference

[MipmapChain Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista



MipmapChain Members

The following tables list the members exposed by the MipmapChain type.


Public Constructors

	Name	Description
	MipmapChain	Overloaded. Initializes a new instance of MipmapChain.



















Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)





Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
	 op_Implicit	Constructs a new mipmap chain containing the specified bitmap.
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	SetItem	Overloaded. Modifies the specified child object in the collection.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ClearItems	(Inherited from Collection .)
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	(Inherited from Collection .)

See Also

Reference

[MipmapChain Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChain Constructor

Initializes a new instance of **MipmapChain**.

Overload List

Name	Description
MipmapChain ()	Initializes a new instance of MipmapChain .
MipmapChain (BitmapContent)	Initializes a new instance of MipmapChain with the specified mipmap.

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChain Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **MipmapChain**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MipmapChain ()
```

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChain Constructor (BitmapContent)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **MipmapChain** with the specified mipmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MipmapChain (  
    BitmapContent bitmap  
)
```

Parameters

bitmap

Image containing a mipmap.

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)





PlatformsWindows XP SP2, Windows Vista

MipmapChain Methods

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
 	Op_Implicit	Constructs a new mipmap chain containing the specified bitmap.
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	SetItem	Overloaded. Modifies the specified child object in the collection.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ClearItems	(Inherited from Collection .)
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	(Inherited from Collection .)

See Also

Reference

[MipmapChain Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChain.InsertItem Method

Inserts a child object into the collection at the specified location.

Overload List

Name	Description
MipmapChain.InsertItem (Int32, BitmapContent)	Inserts a child object into the collection at the specified location.
MipmapChain.InsertItem (Int32, BitmapContent)	(Inherited from Collection .)

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChain.InsertItem Method (Int32, BitmapContent)

Note

This method is available only when developing for Windows.

Inserts a child object into the collection at the specified location.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void InsertItem (  
    int index,  
    BitmapContent item  
)
```

Parameters

index

The position in the collection.

item

The child object being inserted.

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MipmapChain.op_Implicit Method

Constructs a new mipmap chain containing the specified bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static MipmapChain op_Implicit (  
    BitmapContent bitmap  
)
```

Parameters

bitmap

Bitmap used for the mipmap chain.

Return Value

Resultant mipmap chain.

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChain.SetItem Method

Modifies the specified child object in the collection.

Overload List

Name	Description
MipmapChain.SetItem (Int32, BitmapContent)	Modifies the specified child object in the collection.
MipmapChain.SetItem (Int32, BitmapContent)	(Inherited from Collection .)

See Also

Reference

[MipmapChain Class](#)

[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChain.SetItem Method (Int32, BitmapContent)

Note

This method is available only when developing for Windows.

Modifies the specified child object in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void SetItem (  
    int index,  
    BitmapContent item  
)
```

Parameters

index

The position in the collection.

item

The child object being modified.

See Also

Reference

[MipmapChain Class](#)



[MipmapChain Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChain Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[MipmapChain Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChainCollection Class

Note

This class is available only when developing for Windows.

Provides methods for maintaining a mipmap chain.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class MipmapChainCollection : Collection<MipmapChain>
```

Remarks

The collection holds a single mipmap chain for a regular 2D texture, six for a cube map, or an arbitrary number for volume and array textures.

See Also

Reference

[MipmapChainCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

MipmapChainCollection Members

The following tables list the members exposed by the MipmapChainCollection type.

















Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)





Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	SetItem	Overloaded. Modifies the specified child object in the collection.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ClearItems	Removes all children from the collection.
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	Removes a child object in the collection.

See Also

















Reference

[MipmapChainCollection Class](#)





[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChainCollection Methods

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 InsertItem	Overloaded. Inserts a child object into the collection at the specified location.
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 SetItem	Overloaded. Modifies the specified child object in the collection.
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ClearItems	Removes all children from the collection.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	Removes a child object in the collection.

See Also

Reference

[MipmapChainCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChainCollection.ClearItems Method

Note

This method is available only when developing for Windows.

Removes all children from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void ClearItems ()
```

See Also

Reference

[MipmapChainCollection Class](#)

[MipmapChainCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChainCollection.InsertItem Method

Inserts a child object into the collection at the specified location.

Overload List

Name	Description
MipmapChainCollection.InsertItem (Int32, MipmapChain)	Inserts a child object into the collection at the specified location.
MipmapChainCollection.InsertItem (Int32, MipmapChain)	(Inherited from Collection .)

See Also

Reference

[MipmapChainCollection Class](#)

[MipmapChainCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChainCollection.InsertItem Method (Int32, MipmapChain)

Note

This method is available only when developing for Windows.

Inserts a child object into the collection at the specified location.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void InsertItem (  
    int index,  
    MipmapChain item  
)
```

Parameters

index

The position in the collection.

item

The child object being inserted.

See Also

Reference

[MipmapChainCollection Class](#)

[MipmapChainCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChainCollection.RemoveItem Method

Note

This method is available only when developing for Windows.

Removes a child object in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void RemoveItem (  
    int index  
)
```

Parameters

index

The index of the item being removed.

See Also

Reference

[MipmapChainCollection Class](#)

[MipmapChainCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MipmapChainCollection.SetItem Method

Modifies the specified child object in the collection.

Overload List

Name	Description
MipmapChainCollection.SetItem (Int32, MipmapChain)	Modifies the specified child object in the collection.
MipmapChainCollection.SetItem (Int32, MipmapChain)	(Inherited from Collection .)

See Also

Reference

[MipmapChainCollection Class](#)

[MipmapChainCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

MipmapChainCollection.SetItem Method (Int32, MipmapChain)

Note

This method is available only when developing for Windows.

Modifies the specified child object in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void SetItem (  
    int index,  
    MipmapChain item  
)
```

Parameters

index

The position in the collection.

item

The child object being modified.

See Also

Reference

[MipmapChainCollection Class](#)



[MipmapChainCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)


PlatformsWindows XP SP2, Windows Vista

MipmapChainCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[MipmapChainCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

NodeContent Class

Note

This class is available only when developing for Windows.

Provides a base class for graphics types that define local coordinate systems.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class NodeContent : ContentItem
```

Remarks

These objects can be arranged in a tree structure. This enables the root transform to be automatically inherited from the parent object.

See Also

Reference

[NodeContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista









NodeContent Members

The following tables list the members exposed by the NodeContent type.






Public Constructors

Name	Description
 NodeContent	Creates an instance of NodeContent .



Public Properties

Name	Description
 AbsoluteTransform	Gets the value of the local Transform property, multiplied by the AbsoluteTransform of the parent.
 Animations	Gets the set of animations belonging to this node.
 Children	Gets the children of the NodeContent object.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Parent	Gets the parent of this NodeContent object.
 Transform	Gets the transform matrix of the scene.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[NodeContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

NodeContent Constructor

Note

This constructor is available only when developing for Windows.

Creates an instance of [NodeContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public NodeContent ()
```

See Also

Reference

[NodeContent Class](#)






[NodeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

NodeContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[NodeContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

NodeContent Properties

Public Properties

	Name	Description
	AbsoluteTransform	Gets the value of the local Transform property, multiplied by the AbsoluteTransform of the parent.
	Animations	Gets the set of animations belonging to this node.
	Children	Gets the children of the NodeContent object.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Parent	Gets the parent of this NodeContent object.
	Transform	Gets the transform matrix of the scene.

See Also

Reference

[NodeContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

NodeContent.AbsoluteTransform Property

Note

This property is available only when developing for Windows.

Gets the value of the local [Transform](#) property, multiplied by the [AbsoluteTransform](#) of the parent.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Matrix AbsoluteTransform { get; }
```

Property Value

Matrix of the [NodeContent](#) object.

See Also

Reference

[NodeContent Class](#)

[NodeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NodeContent.Animations Property

Note

This property is available only when developing for Windows.

Gets the set of animations belonging to this node.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public AnimationContentDictionary Animations { get; }
```

Property Value

Collection of animations for this content item.

Remarks

Note

The XNA Framework provides only partial support for animation. In the [NodeContent](#) class an intermediate object model for storing animation data inside the Content Pipeline is defined, which exposes data into that has been extracted from Autodesk FBX (.fbx) and DirectX X file (.x) formats. The built-in [ModelProcessor](#) also converts vertex channels of [BoneWeightCollection](#) data into sets of [VertexElementUsage.BlendIndices](#) and [VertexElementUsage.BlendWeight](#) channels suitable for skinned rendering on the GPU. However, the framework does not include any runtime animation classes at this time.

There can be any number of animations, which are looked up by name. For character skinning, the animation data is usually attached to the root of a bone hierarchy. However, it can potentially belong to any node. For example, when used for rigid body animation.

See Also

Reference

[NodeContent Class](#)

[NodeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

NodeContent.Children Property

Note

This property is available only when developing for Windows.

Gets the children of the [NodeContent](#) object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public NodeContentCollection Children { get; }
```

Property Value

Collection of children.

See Also

Reference

[NodeContent Class](#)

[NodeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NodeContent.Parent Property

Note

This property is available only when developing for Windows.

Gets the parent of this [NodeContent](#) object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public NodeContent Parent { get; set; }
```

Property Value

Parent of the [NodeContent](#) object, or null if this object is the root of the scene.

See Also

Reference

[NodeContent Class](#)

[NodeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NodeContent.Transform Property

Note

This property is available only when developing for Windows.

Gets the transform matrix of the scene.

The transform matrix defines a local coordinate system for the content in addition to any children of this object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Matrix Transform { get; set; }
```

Property Value

Transform matrix used by the scene.

See Also

Reference

[NodeContent Class](#)

[NodeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NodeContentCollection Class

Note

This class is available only when developing for Windows.

Provides a collection of all [NodeContent](#) objects in a spatial hierarchy.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class NodeContentCollection : ChildCollection<NodeContent, NodeContent>
```

See Also

Reference

[NodeContentCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

NodeContentCollection Members

The following tables list the members exposed by the NodeContentCollection type.

















Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)





Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	InsertItem	(Inherited from Collection .)
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	SetItem	(Inherited from Collection .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	GetParent	Gets the parent of a child object.
	MemberwiseClone	(Inherited from Object .)
	SetParent	Gets the parent of a child object.

See Also

















Reference

[NodeContentCollection Class](#)





[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

NodeContentCollection Methods

Public Methods

Name	Description
 Add	(Inherited from Collection .)
 Clear	(Inherited from Collection .)
 Contains	(Inherited from Collection .)
 CopyTo	(Inherited from Collection .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from Collection .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	(Inherited from Collection .)
 Insert	(Inherited from Collection .)
 InsertItem	(Inherited from Collection .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from Collection .)
 RemoveAt	(Inherited from Collection .)
 SetItem	(Inherited from Collection .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetParent	Gets the parent of a child object.
 MemberwiseClone	(Inherited from Object .)
 SetParent	Gets the parent of a child object.

See Also

Reference

[NodeContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

NodeContentCollection.GetParent Method

Note

This method is available only when developing for Windows.

Gets the parent of a child object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override NodeContent GetParent (  
    NodeContent child  
)
```

Parameters

child

The child of the parent being retrieved.

Return Value

The parent of the child object.

See Also

Reference

[NodeContentCollection Class](#)

[NodeContentCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

NodeContentCollection.SetParent Method

Note

This method is available only when developing for Windows.

Gets the parent of a child object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void SetParent (  
    NodeContent child,  
    NodeContent parent  
)
```

Parameters

child

The child of the parent being retrieved.

parent

The parent of the child object.

See Also

Reference

[NodeContentCollection Class](#)



[NodeContentCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)


Platforms Windows XP SP2, Windows Vista

NodeContentCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[NodeContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PixelBitmapContent Generic Class

Note

This generic class is available only when developing for Windows.

Provides methods for maintaining a 2D array of pixel values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class PixelBitmapContent<T> : BitmapContent where T : ValueType, IEquatable
```

Remarks

Pixel bitmaps are represented as a 2D array of independent pixel values. The pixel type will often be from the [Graphics.PackedVector](#) namespace, but can also be [System.Single](#), [Vector2](#), [Vector3](#), or [Vector4](#). A pixel bitmap of [Vector4](#) values is commonly used as an intermediate format for image manipulation operations.

See Also

Reference

[PixelBitmapContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista






PixelBitmapContent Members

The following tables list the members exposed by the PixelBitmapContent type.














Public Constructors

Name	Description
 PixelBitmapContent	Overloaded. Initializes a new instance of PixelBitmapContent.






Public Properties

Name	Description
 Height	(Inherited from BitmapContent .)
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)
 Width	(Inherited from BitmapContent .)

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetPixel	Gets the pixel value at the specified location in the bitmap.
 GetPixelData	Gets the bitmap content as an array of encoded bytes.
 GetRow	Gets an array containing a single row of bitmap data.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ReplaceColor	Searches for all pixels matching the specified color and replaces them with a new value.
 SetPixel	Sets the pixel value at the specified location in the bitmap.
 SetPixelData	Sets the bitmap content using the specified encoded byte array.
 ToString	Gets a string description of the bitmap.
 TryGetFormat	Gets the GPU texture format that corresponds to the specified bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 TryCopyFrom	Attempts to copy a region from the specified bitmap.
 TryCopyTo	Attempts to copy a region to the specified bitmap.
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[PixelBitmapContent Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PixelBitmapContent Constructor

Initializes a new instance of **PixelBitmapContent**.

Overload List

Name	Description
PixelBitmapContent ()	Initializes a new instance of PixelBitmapContent .
PixelBitmapContent (Int32, Int32)	Initializes a new instance of PixelBitmapContent with the specified width and height.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PixelBitmapContent Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **PixelBitmapContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected PixelBitmapContent ()
```

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent Constructor (Int32, Int32)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **PixelBitmapContent** with the specified width and height.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PixelBitmapContent (  
    int width,  
    int height  
)
```

Parameters

width

Width, in pixels, of the blank pixel bitmap.

height

Height, in pixels, of the blank pixel bitmap.

See Also

Reference

[PixelBitmapContent Generic Class](#)














[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)






PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent Methods

Public Methods

Name	Description
 Copy	(Inherited from BitmapContent .)
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetPixel	Gets the pixel value at the specified location in the bitmap.
 GetPixelData	Gets the bitmap content as an array of encoded bytes.
 GetRow	Gets an array containing a single row of bitmap data.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ReplaceColor	Searches for all pixels matching the specified color and replaces them with a new value.
 SetPixel	Sets the pixel value at the specified location in the bitmap.
 SetPixelData	Sets the bitmap content using the specified encoded byte array.
 ToString	Gets a string description of the bitmap.
 TryGetFormat	Gets the GPU texture format that corresponds to the specified bitmap type.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 TryCopyFrom	Attempts to copy a region from the specified bitmap.
 TryCopyTo	Attempts to copy a region to the specified bitmap.
 ValidateCopyArguments	(Inherited from BitmapContent .)

See Also

Reference

[PixelBitmapContent Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PixelBitmapContent.GetPixel Method

Note

This method is available only when developing for Windows.

Gets the pixel value at the specified location in the bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T GetPixel (  
    int x,  
    int y  
)
```

Parameters

x

The x component of a bitmap pixel.

y

The y component of a bitmap pixel.

Return Value

Value of the pixel.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent.GetPixelData Method

Note

This method is available only when developing for Windows.

Gets the bitmap content as an array of encoded bytes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override byte[] GetPixelData ()
```

Return Value

Contents of the bitmap.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent.GetRow Method

Note

This method is available only when developing for Windows.

Gets an array containing a single row of bitmap data.

This method can be used both to read and modify the bitmap contents.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T[] GetRow (  
    int y  
)
```

Parameters

y

The y component of the bitmap row.

Return Value

Array containing the bitmap row data.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent.ReplaceColor Method

Note

This method is available only when developing for Windows.

Searches for all pixels matching the specified color and replaces them with a new value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void ReplaceColor (  
    T originalColor,  
    T newColor  
)
```

Parameters

originalColor

Color being searched for.

newColor

Color used to replace *originalColor*.

Remarks

Use this method to implement color keying. For instance, replace [Magenta](#) with [TransparentBlack](#).

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent.SetPixel Method

Note

This method is available only when developing for Windows.

Sets the pixel value at the specified location in the bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void SetPixel (  
    int x,  
    int y,  
    T value  
)
```

Parameters

x

The x component of a bitmap pixel.

y

The y component of a bitmap pixel.

value

New pixel value.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent.SetPixelData Method

Note

This method is available only when developing for Windows.

Sets the bitmap content using the specified encoded byte array.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override void SetPixelData (  
    byte[] sourceData  
)
```

Parameters

sourceData

New bitmap content.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

PixelBitmapContent.ToString Method

Note

This method is available only when developing for Windows.

Gets a string description of the bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override string ToString ()
```

Return Value

Description of the bitmap.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PixelBitmapContent.TryCopyFrom Method

Note

This method is available only when developing for Windows.

Attempts to copy a region from the specified bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override bool TryCopyFrom (  
    BitmapContent sourceBitmap,  
    Rectangle sourceRegion,  
    Rectangle destinationRegion  
)
```

Parameters

sourceBitmap

Bitmap content being copied.

sourceRegion

Location of the upper-left region of *sourceBitmap* being copied from.

destinationRegion

Region of destination bitmap to be overwritten.

Return Value

true if region copy is supported; **false** otherwise.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

PixelBitmapContent.TryCopyTo Method

Note

This method is available only when developing for Windows.

Attempts to copy a region to the specified bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override bool TryCopyTo (  
    BitmapContent destinationBitmap,  
    Rectangle sourceRegion,  
    Rectangle destinationRegion  
)
```

Parameters

destinationBitmap

Bitmap to be copied to.

sourceRegion

Location of upper-left region to be copied.

destinationRegion

Target region for the copied content.

Return Value

true if region copy is supported; **false** otherwise.

See Also

Reference

[PixelBitmapContent Generic Class](#)

[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

PixelBitmapContent.TryGetFormat Method

Note

This method is available only when developing for Windows.

Gets the GPU texture format that corresponds to the specified bitmap type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool TryGetFormat (  
    out SurfaceFormat format  
)
```

Parameters

format

[[OutAttribute](#)] Format being compared.

Return Value

true if the bitmap matches *format*; **false** otherwise.

See Also

Reference

[PixelBitmapContent Generic Class](#)






[PixelBitmapContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

PixelBitmapContent Properties

Public Properties

	Name	Description
	Height	(Inherited from BitmapContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	Width	(Inherited from BitmapContent .)

See Also

Reference

[PixelBitmapContent Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PositionCollection Class

Note

This class is available only when developing for Windows.

Provides a collection of vertex position values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class PositionCollection : Collection<Vector3>
```

See Also

Reference

[PositionCollection Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



PositionCollection Members

The following tables list the members exposed by the PositionCollection type.


Public Constructors

	Name	Description
	PositionCollection	Initializes a new instance of PositionCollection.















Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)







Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

Public Methods

	Name	Description
	Add	(Inherited from Collection .)
	Clear	(Inherited from Collection .)
	Contains	(Inherited from Collection .)
	CopyTo	(Inherited from Collection .)
	Equals	(Inherited from Object .)
	GetEnumerator	(Inherited from Collection .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	(Inherited from Collection .)
	Insert	(Inherited from Collection .)
	ReferenceEquals	(Inherited from Object .)
	Remove	(Inherited from Collection .)
	RemoveAt	(Inherited from Collection .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ClearItems	(Inherited from Collection .)
	Finalize	(Inherited from Object .)
	InsertItem	(Inherited from Collection .)
	MemberwiseClone	(Inherited from Object .)
	RemoveItem	(Inherited from Collection .)
	SetItem	(Inherited from Collection .)

See Also

Reference

[PositionCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PositionCollection Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **PositionCollection**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PositionCollection ()
```

See Also

Reference

[PositionCollection Class](#)















[PositionCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)







PlatformsWindows XP SP2, Windows Vista

PositionCollection Methods

Public Methods

Name	Description
 Add	(Inherited from Collection.)
 Clear	(Inherited from Collection.)
 Contains	(Inherited from Collection.)
 CopyTo	(Inherited from Collection.)
 Equals	(Inherited from Object.)
 GetEnumerator	(Inherited from Collection.)
 GetHashCode	(Inherited from Object.)
 GetType	(Inherited from Object.)
 IndexOf	(Inherited from Collection.)
 Insert	(Inherited from Collection.)
 ReferenceEquals	(Inherited from Object.)
 Remove	(Inherited from Collection.)
 RemoveAt	(Inherited from Collection.)
 ToString	(Inherited from Object.)

Protected Methods

Name	Description
 ClearItems	(Inherited from Collection.)
 Finalize	(Inherited from Object.)
 InsertItem	(Inherited from Collection.)
 MemberwiseClone	(Inherited from Object.)
 RemoveItem	(Inherited from Collection.)
 SetItem	(Inherited from Collection.)

See Also



Reference

[PositionCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PositionCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from Collection .)
	Item	(Inherited from Collection .)

Protected Properties

	Name	Description
	Items	(Inherited from Collection .)

See Also

Reference

[PositionCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture2DContent Class

Note

This class is available only when developing for Windows.

Represents a regular two-dimensional texture.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class Texture2DContent : TextureContent
```

See Also

Reference

[Texture2DContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista






Texture2DContent Members

The following tables list the members exposed by the Texture2DContent type.









Public Constructors

Name	Description
 Texture2DContent	Initializes a new instance of Texture2DContent.



Public Properties

Name	Description
 Faces	(Inherited from TextureContent .)
 Identity	(Inherited from ContentItem .)
 Mipmaps	Gets or sets the collection of mipmaps containing the texture image data.
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)

Public Methods

Name	Description
 ConvertBitmapType	(Inherited from TextureContent .)
 Equals	(Inherited from Object .)
 GenerateMipmaps	(Inherited from TextureContent .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Validate	(Inherited from TextureContent .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Texture2DContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture2DContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **Texture2DContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Texture2DContent ()
```

See Also

Reference

[Texture2DContent Class](#)









[Texture2DContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

Texture2DContent Methods

Public Methods

	Name	Description
	ConvertBitmapType	(Inherited from TextureContent .)
	Equals	(Inherited from Object .)
	GenerateMipmaps	(Inherited from TextureContent .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	(Inherited from TextureContent .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[Texture2DContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture2DContent Properties

Public Properties

	Name	Description
	Faces	(Inherited from TextureContent .)
	Identity	(Inherited from ContentItem .)
	Mipmaps	Gets or sets the collection of mipmaps containing the texture image data.
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[Texture2DContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture2DContent.Mipmaps Property

Note

This property is available only when developing for Windows.

Gets or sets the collection of mipmaps containing the texture image data.

This is a shortcut for accessing **base.Faces[0]**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MipmapChain Mipmaps { get; set; }
```

Property Value

Mipmap chain containing the texture images of an object.

See Also

Reference

[Texture2DContent Class](#)

[Texture2DContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Texture3DContent Class

Note

This class is available only when developing for Windows.

Represents a three-dimensional volume texture.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class Texture3DContent : TextureContent
```

See Also

Reference

[Texture3DContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista





Texture3DContent Members

The following tables list the members exposed by the Texture3DContent type.









Public Constructors

Name	Description
 Texture3DContent	Initializes a new instance of Texture3DContent.



Public Properties

Name	Description
 Faces	(Inherited from TextureContent .)
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)

Public Methods

Name	Description
 ConvertBitmapType	(Inherited from TextureContent .)
 Equals	(Inherited from Object .)
 GenerateMipmaps	Generates mipmaps for a volume texture.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Validate	Overloaded. Validates that the required contents of this volume texture are present and valid.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Texture3DContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture3DContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **Texture3DContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Texture3DContent ()
```

See Also

Reference

[Texture3DContent Class](#)









[Texture3DContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

Texture3DContent Methods

Public Methods

	Name	Description
	ConvertBitmapType	(Inherited from TextureContent .)
	Equals	(Inherited from Object .)
	GenerateMipmaps	Generates mipmaps for a volume texture.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	Overloaded. Validates that the required contents of this volume texture are present and valid.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[Texture3DContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture3DContent.GenerateMipmaps Method

Note

This method is available only when developing for Windows.

Generates mipmaps for a volume texture.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override void GenerateMipmaps (  
    bool overwriteExistingMipmaps  
)
```

Parameters

overwriteExistingMipmaps

true to overwrite existing mipmaps for the volume texture; **false** otherwise.

See Also

Reference

[Texture3DContent Class](#)

[Texture3DContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture3DContent.Validate Method

Validates that the required contents of this volume texture are present and valid.

Overload List

Name	Description
Texture3DContent.Validate ()	Validates that the required contents of this volume texture are present and valid. This method verifies that mip counts are correct down each individual 2D face, and that width and height are shrinking appropriately.
Texture3DContent.Validate (Boolean)	(Inherited from TextureContent .)

See Also

Reference

[Texture3DContent Class](#)

[Texture3DContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Texture3DContent.Validate Method ()

Note

This method is available only when developing for Windows.

Validates that the required contents of this volume texture are present and valid. This method verifies that mip counts are correct down each individual 2D face, and that width and height are shrinking appropriately.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override void Validate ()
```

Exceptions

Exception type	Condition
InvalidContentException	The texture is not a valid one.

See Also

Reference

[Texture3DContent Class](#)





[Texture3DContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

Texture3DContent Properties

Public Properties

	Name	Description
	Faces	(Inherited from TextureContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[Texture3DContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureContent Class

Note

This class is available only when developing for Windows.

Provides a base class for all texture objects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class TextureContent : ContentItem
```

See Also

Reference

[TextureContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista





TextureContent Members

The following tables list the members exposed by the TextureContent type.









Protected Constructors

Name	Description
 TextureContent	Initializes a new instance of TextureContent with the specified face collection.



Public Properties

Name	Description
 Faces	Collection of image faces that hold a single mipmap chain for a regular 2D texture, six chains for a cube map, or an arbitrary number for volume and array textures.
 Identity	(Inherited from ContentItem .)
 Name	(Inherited from ContentItem .)
 OpaqueData	(Inherited from ContentItem .)

Public Methods

Name	Description
 ConvertBitmapType	Converts all bitmaps for this texture to a different format.
 Equals	(Inherited from Object .)
 GenerateMipmaps	Generates a full set of mipmaps for the texture.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Validate	Overloaded. Verifies that all contents of this texture are present and correct.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **TextureContent** with the specified face collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected TextureContent (  
    MipmapChainCollection faces  
)
```

Parameters

faces

Mipmap chain containing the face collection.

Remarks

Override this constructor to provide support for specialized face collections. For example, the [cube map](#) class provides a fixed-size collection of six faces, while the [Texture2DContent](#) class provides a variable-size collection.

See Also

Reference

[TextureContent Class](#)









[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



PlatformsWindows XP SP2, Windows Vista

TextureContent Methods

Public Methods

	Name	Description
	ConvertBitmapType	Converts all bitmaps for this texture to a different format.
	Equals	(Inherited from Object .)
	GenerateMipmaps	Generates a full set of mipmaps for the texture.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	Overloaded. Verifies that all contents of this texture are present and correct.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureContent.ConvertBitmapType Method

Note

This method is available only when developing for Windows.

Converts all bitmaps for this texture to a different format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void ConvertBitmapType (
    Type newBitmapType
)
```

Parameters

newBitmapType

Type being converted to. The new type must be a subclass of [BitmapContent](#), such as [PixelBitmapContent](#) or [DxtBitmapContent](#).

Exceptions

Exception type	Condition
ArgumentException	<ul style="list-style-type: none"> ConvertBitmapType cannot convert to <i>newBitmapType</i>. The target type must be derived from BitmapContent. ConvertBitmapType cannot convert to <i>newBitmapType</i>. The target must be a concrete type with a public constructor taking integer width and height parameters.
ArgumentNullException	<i>newBitmapType</i> is null .

See Also

Reference

[TextureContent Class](#)

[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureContent.GenerateMipmaps Method

Note

This method is available only when developing for Windows.

Generates a full set of mipmaps for the texture.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual void GenerateMipmaps (  
    bool overwriteExistingMipmaps  
)
```

Parameters

overwriteExistingMipmaps

true if the existing mipmap set is replaced with the new set; **false** otherwise.

See Also

Reference

[TextureContent Class](#)

[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureContent.Validate Method

Verifies that all contents of this texture are present and correct.

Overload List

Name	Description
TextureContent.Validate ()	Verifies that all contents of this texture are present and correct.
TextureContent.Validate (Boolean)	Verifies that all contents of this texture are present and correct.

See Also

Reference

[TextureContent Class](#)

[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureContent.Validate Method ()

Note

This method is available only when developing for Windows.

Verifies that all contents of this texture are present and correct.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual void Validate ()
```

Remarks

If texture verification fails, [InvalidContentException](#) is thrown. The base implementation of this method calls [Validate](#) with a value of **true**.

See Also

Reference

[TextureContent Class](#)

[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureContent.Validate Method (Boolean)

Note

This method is available only when developing for Windows.

Verifies that all contents of this texture are present and correct.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected void Validate (  
    bool facesMustHaveSameMipCount  
)
```

Parameters

facesMustHaveSameMipCount

Result of the validation.

Remarks

Validation is based on the following rules.

- One image face must be present.
- The mipmap chain for each face contains at least one bitmap.
- All bitmaps share the same type.
- Each face is the same size.
- Each face contains the same number of mipmaps.
- Each mipmap must be half the size of the previous. If the root bitmap is not square, the smaller axis is rounded up to 1 for the last few mipmaps before both dimensions reach zero.

Derived validation methods could add more validation rules. For example, cube maps must be square.

Exceptions

Exception type	Condition
InvalidContentException	The texture is not a valid one.

See Also

Reference

[TextureContent Class](#)





[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureContent Properties

Public Properties

	Name	Description
	Faces	Collection of image faces that hold a single mipmap chain for a regular 2D texture, six chains for a cube map, or an arbitrary number for volume and array textures.
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	Opaque Data	(Inherited from ContentItem .)

See Also

Reference

[TextureContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureContent.Faces Property

Note

This property is available only when developing for Windows.

Collection of image faces that hold a single mipmap chain for a regular 2D texture, six chains for a cube map, or an arbitrary number for volume and array textures.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MipmapChainCollection Faces { get; }
```

Property Value

Collection of object faces.

See Also

Reference

[TextureContent Class](#)

[TextureContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureCubeContent Class

Note

This class is available only when developing for Windows.

Provides validation for a cube map texture, which contains an array of six image faces.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class TextureCubeContent : TextureContent
```

See Also

Reference

[TextureCubeContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista





TextureCubeContent Members

The following tables list the members exposed by the TextureCubeContent type.









Public Constructors

	Name	Description
	TextureCubeContent	Initializes a new instance of TextureCubeContent.



Public Properties

	Name	Description
	Faces	(Inherited from TextureContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

Public Methods

	Name	Description
	ConvertBitmapType	(Inherited from TextureContent .)
	Equals	(Inherited from Object .)
	GenerateMipmaps	(Inherited from TextureContent .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	Overloaded. Verifies that all contents of this texture are present and correct.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureCubeContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureCubeContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **TextureCubeContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TextureCubeContent ()
```

See Also

Reference

[TextureCubeContent Class](#)









[TextureCubeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



Platforms Windows XP SP2, Windows Vista

TextureCubeContent Methods

Public Methods

	Name	Description
	ConvertBitmapType	(Inherited from TextureContent .)
	Equals	(Inherited from Object .)
	GenerateMipmaps	(Inherited from TextureContent .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Validate	Overloaded. Verifies that all contents of this texture are present and correct.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureCubeContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureCubeContent.Validate Method

Verifies that all contents of this texture are present and correct.

Overload List

Name	Description
TextureCubeContent.Validate ()	Verifies that all contents of the cube map are present and correct.
TextureCubeContent.Validate (Boolean)	(Inherited from TextureContent.)

See Also

Reference

[TextureCubeContent Class](#)

[TextureCubeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureCubeContent.Validate Method ()

Note

This method is available only when developing for Windows.

Verifies that all contents of the cube map are present and correct.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override void Validate ()
```

Exceptions

Exception type	Condition
InvalidContentException	The cube map texture is not a valid one.

See Also

Reference

[TextureCubeContent Class](#)





[TextureCubeContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureCubeContent Properties

Public Properties

	Name	Description
	Faces	(Inherited from TextureContent .)
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)

See Also

Reference

[TextureCubeContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureReferenceDictionary Class

Note

This class is available only when developing for Windows.

Provides a collection of named references to texture files.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class TextureReferenceDictionary : NamedValueDictionary<ExternalReference<TextureContent>>
```

See Also

Reference

[TextureReferenceDictionary Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista





TextureReferenceDictionary Members

The following tables list the members exposed by the TextureReferenceDictionary type.












Public Constructors

Name	Description
 TextureReferenceDictionary	Initializes a new instance of TextureReferenceDictionary.







Public Properties

Name	Description
 Count	(Inherited from NamedValueDictionary .)
 Item	(Inherited from NamedValueDictionary .)
 Keys	(Inherited from NamedValueDictionary .)
 Values	(Inherited from NamedValueDictionary .)

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 AddItem	(Inherited from NamedValueDictionary .)
 ClearItems	(Inherited from NamedValueDictionary .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from NamedValueDictionary .)
 SetItem	(Inherited from NamedValueDictionary .)

See Also

Reference

[TextureReferenceDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureReferenceDictionary Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **TextureReferenceDictionary**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TextureReferenceDictionary ()
```

See Also

Reference

[TextureReferenceDictionary Class](#)












[TextureReferenceDictionary Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)







Platforms Windows XP SP2, Windows Vista

TextureReferenceDictionary Methods

Public Methods

Name	Description
 Add	(Inherited from NamedValueDictionary .)
 Clear	(Inherited from NamedValueDictionary .)
 ContainsKey	(Inherited from NamedValueDictionary .)
 Equals	(Inherited from Object .)
 GetEnumerator	(Inherited from NamedValueDictionary .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Remove	(Inherited from NamedValueDictionary .)
 ToString	(Inherited from Object .)
 TryGetValue	(Inherited from NamedValueDictionary .)

Protected Methods

Name	Description
 AddItem	(Inherited from NamedValueDictionary .)
 ClearItems	(Inherited from NamedValueDictionary .)
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 RemoveItem	(Inherited from NamedValueDictionary .)
 SetItem	(Inherited from NamedValueDictionary .)

See Also





Reference

[TextureReferenceDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

TextureReferenceDictionary Properties

Public Properties

	Name	Description
	Count	(Inherited from NamedValueDictionary.)
	Item	(Inherited from NamedValueDictionary.)
	Keys	(Inherited from NamedValueDictionary.)
	Values	(Inherited from NamedValueDictionary.)

See Also

Reference

[TextureReferenceDictionary Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VectorConverter Class

Note

This class is available only when developing for Windows.

Provides methods for converting data between different packed vector representations.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static class VectorConverter
```

See Also

Reference

[VectorConverter Members](#)














[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista



VectorConverter Members

The following tables list the members exposed by the VectorConverter type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
 	GetConverter	Gets a Converter delegate that can convert one packed vector representation to another.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
 	TryGetSurfaceFormat	Gets the GPU texture format that corresponds to a managed vector type.
 	TryGetVectorType	Overloaded. Searches for a managed vector type that corresponds to the specified format type.
 	TryGetVertexElementFormat	Gets the GPU vertex format that corresponds to a managed vector type.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also










Reference

[VectorConverter Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VectorConverter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetConverter	Gets a Converter delegate that can convert one packed vector representation to another.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	TryGetSurfaceFormat	Gets the GPU texture format that corresponds to a managed vector type.
	TryGetVectorType	Overloaded. Searches for a managed vector type that corresponds to the specified format type.
	TryGetVertexElementFormat	Gets the GPU vertex format that corresponds to a managed vector type.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VectorConverter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VectorConverter.GetConverter Generic Method

Note

This generic method is available only when developing for Windows.

Gets a [Converter](#) delegate that can convert one packed vector representation to another.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static Converter<TInput, TOutput> GetConverter<TInput,TOutput> ()
```

Type Parameters

TInput

Source type for the conversion.

TOutput

Output type for the conversion.

Return Value

A [Converter](#) delegate that encodes or decodes *TInput* to *TOutput*.

Remarks

The input and output types must be either a value type that implements [IPackedVector](#) or one of [Single](#), [Vector2](#), [Vector3](#), or [Vector4](#).

See Also

Reference

[VectorConverter Class](#)

[VectorConverter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VectorConverter.TryGetSurfaceFormat Method

Note

This method is available only when developing for Windows.

Gets the GPU texture format that corresponds to a managed vector type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static bool TryGetSurfaceFormat (  
    Type vectorType,  
    out SurfaceFormat surfaceFormat  
)
```

Parameters

vectorType

Can be one of the following:

- [IPackedVector](#)
- [Single](#)
- [Vector2](#)
- [Vector3](#)
- [Vector4](#)

surfaceFormat

[[OutAttribute](#)] Surface format being compared.

Return Value

true if a match is found; **false** otherwise.

See Also

Reference

[VectorConverter Class](#)

[VectorConverter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VectorConverter.TryGetVectorType Method

Searches for a managed vector type that corresponds to the specified format type.

Overload List

Name	Description
VectorConverter.TryGetVectorType (SurfaceFormat, Type)	Searches for a managed vector type that corresponds to a GPU texture format.
VectorConverter.TryGetVectorType (VertexElementFormat, Type)	Searches for a managed vector type that corresponds to a GPU vertex format.

See Also

Reference

[VectorConverter Class](#)

[VectorConverter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VectorConverter.TryGetVectorType Method (SurfaceFormat, Type)

Note

This method is available only when developing for Windows.

Searches for a managed vector type that corresponds to a GPU texture format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static bool TryGetVectorType (  
    SurfaceFormat surfaceFormat,  
    out Type vectorType  
)
```

Parameters

surfaceFormat

Surface format being compared.

vectorType

[[OutAttribute](#)] Corresponding managed vector type.

Return Value

true if a match is found; **false** otherwise.

See Also

Reference

[VectorConverter Class](#)

[VectorConverter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VectorConverter.TryGetVectorType Method (VertexElementFormat, Type)

Note

This method is available only when developing for Windows.

Searches for a managed vector type that corresponds to a GPU vertex format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static bool TryGetVectorType (  
    VertexElementFormat vertexElementFormat,  
    out Type vectorType  
)
```

Parameters

vertexElementFormat

Texture format being matched.

vectorType

[[OutAttribute](#)] Corresponding managed vector type.

Return Value

true if a match is found; **false** otherwise.

See Also

Reference

[VectorConverter Class](#)

[VectorConverter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VectorConverter.TryGetVertexElementFormat Method

Note

This method is available only when developing for Windows.

Gets the GPU vertex format that corresponds to a managed vector type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static bool TryGetVertexElementFormat (  
    Type vectorType,  
    out VertexElementFormat vertexElementFormat  
)
```

Parameters

vectorType

Can be one of the following:

- [IPackedVector](#)
- [Single](#)
- [Vector2](#)
- [Vector3](#)
- [Vector4](#)

vertexElementFormat

[[OutAttribute](#)] Corresponding vertex element type.

Return Value

true if a match is found; **false** otherwise.

See Also

Reference

[VectorConverter Class](#)

[VectorConverter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel Class

Note

This class is available only when developing for Windows.

Provides methods and properties for maintaining a vertex channel.

A vertex channel is a list of arbitrary data with one value for each vertex. Channels are stored inside a [GeometryContent](#) and identified by name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class VertexChannel : IList, ICollection, IEnumerable
```

See Also

Reference

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)





[VertexChannelNames](#)

PlatformsWindows XP SP2, Windows Vista











VertexChannel Members

The following tables list the members exposed by the VertexChannel type.



Public Properties

Name	Description
 Count	Gets the number of elements in the vertex channel.
 ElementType	Gets the type of data contained in this channel.
 Item	Gets or sets the element at the specified index.
 Name	Gets the name of the vertex channel.










Public Methods

Name	Description
 Contains	Determines whether the specified element is in the channel.
 CopyTo	Copies the elements of the channel to an array, starting at the specified index.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator interface for reading channel content.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Gets the index of the specified item.
 ReadConvertedContent	Reads channel content and automatically converts it to the specified vector format.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IList.IsFixedSize	Gets a value indicating whether this list has a fixed size.
 System.Collections.IList.IsReadOnly	Gets a value indicating whether this object is read-only.
 System.Collections.ICollection.IsSynchronized	Gets a value indicating whether access to the collection is synchronized (thread safe).
 System.Collections.ICollection.SyncRoot	Gets an object that can be used to synchronize access to the collection.
 System.Collections.IList.Add	Adds a new element to the end of the collection.
 System.Collections.IList.Clear	Removes all elements from the collection.
 System.Collections.IList.Insert	Inserts an element into the collection at the specified position.
 System.Collections.IList.RemoveAt	Removes the element at the specified index position.
 System.Collections.IList.Remove	Removes a specified element from the collection.

See Also











Reference

[VertexChannel Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel Methods










Public Methods

Name	Description
 Contains	Determines whether the specified element is in the channel.
 CopyTo	Copies the elements of the channel to an array, starting at the specified index.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator interface for reading channel content.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Gets the index of the specified item.
 ReadConvertedContent	Reads channel content and automatically converts it to the specified vector format.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.IList.IsFixedSize	Gets a value indicating whether this list has a fixed size.
 System.Collections.IList.IsReadOnly	Gets a value indicating whether this object is read-only.
 System.Collections.ICollection.IsSynchronized	Gets a value indicating whether access to the collection is synchronized (thread safe).
 System.Collections.ICollection.SyncRoot	Gets an object that can be used to synchronize access to the collection.
 System.Collections.IList.Add	Adds a new element to the end of the collection.
 System.Collections.IList.Clear	Removes all elements from the collection.
 System.Collections.IList.Insert	Inserts an element into the collection at the specified position.
 System.Collections.IList.RemoveAt	Removes the element at the specified index position.
 System.Collections.IList.Remove	Removes a specified element from the collection.

See Also

Reference

[VertexChannel Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.Contains Method

Note

This method is available only when developing for Windows.

Determines whether the specified element is in the channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Contains (  
    Object value  
)
```

Parameters

value

Element being searched for.

Return Value

true if the element is present; **false** otherwise.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel.CopyTo Method

Note

This method is available only when developing for Windows.

Copies the elements of the channel to an array, starting at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void CopyTo (  
    Array array,  
    int index  
)
```

Parameters

array

Array that will receive the copied channel elements.

index

Starting index for copy operation.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel.GetEnumerator Method

Note

This method is available only when developing for Windows.

Gets an enumerator interface for reading channel content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerator GetEnumerator ()
```

Return Value

Enumeration of the channel content.

Remarks

To access strongly typed content data, cast the [VertexChannel](#) object to a [VertexChannel<T>](#) before calling **GetEnumerator**. This causes a generic enumerator interface to be returned.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel.IndexOf Method

Note

This method is available only when developing for Windows.

Gets the index of the specified item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int IndexOf (
    Object value
)
```

Parameters

value

Item whose index is to be retrieved.

Return Value

Index of specified item.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.ReadConvertedContent Generic Method

Note

This generic method is available only when developing for Windows.

Reads channel content and automatically converts it to the specified vector format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract IEnumerable<TargetType> ReadConvertedContent<TargetType> ()
```

Type Parameters

TargetType

Return Value

Vector format to convert returned data to.

Remarks

The following formats are supported:

- [Single](#)
- [Vector2 Structure](#)
- [Vector3 Structure](#)
- [Vector4 Structure](#)
- Any implementation of [IPackedVector Interface](#).

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.IList.Add Method

Note

This method is available only when developing for Windows.

Adds a new element to the end of the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private int System.Collections.IList.Add (  
    Object value  
)
```

Parameters

value

The element to add.

Return Value

Index of the element.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.IList.Clear Method

Note

This method is available only when developing for Windows.

Removes all elements from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.IList.Clear ()
```

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.IList.Insert Method

Note

This method is available only when developing for Windows.

Inserts an element into the collection at the specified position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.IList.Insert (  
    int index,  
    Object value  
)
```

Parameters

index

Index at which to insert the element.

value

The element to insert.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.IList.Remove Method

Note

This method is available only when developing for Windows.

Removes a specified element from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.IList.Remove (  
    Object value  
)
```

Parameters

value

The element to remove.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.IList.RemoveAt Method

Note

This method is available only when developing for Windows.

Removes the element at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.IList.RemoveAt (  
    int index  
)
```

Parameters

index

Index of the element to remove.

See Also

Reference

[VertexChannel Class](#)





[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel Properties

Public Properties

	Name	Description
	Count	Gets the number of elements in the vertex channel.
	ElementType	Gets the type of data contained in this channel.
	Item	Gets or sets the element at the specified index.
	Name	Gets the name of the vertex channel.

See Also

Reference

[VertexChannel Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.Count Property

Note

This property is available only when developing for Windows.

Gets the number of elements in the vertex channel

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of elements.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.ElementType Property

Note

This property is available only when developing for Windows.

Gets the type of data contained in this channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract Type ElementType { get; }
```

Property Value

Type of data stored in the vertex channel.

Remarks

To access strongly typed content data, cast the [VertexChannel](#) object to a [VertexChannel<T>](#) before calling **ElementType**. This causes a generic enumerator interface to be returned.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel.Item Property

Note

This property is available only when developing for Windows.

Gets or sets the element at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object this [
    int index
] { get; set; }
```

Property Value

Element at the specified index.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.Name Property

Note

This property is available only when developing for Windows.

Gets the name of the vertex channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

Name of the channel.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.System.Collections.ICollection.IsSynchronized Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether access to the collection is synchronized (thread safe).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.ICollection.IsSynchronized { get; }
```

Property Value

true if access to the collection is synchronized (thread safe); **false** otherwise.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.System.Collections.ICollection.SyncRoot Property

Note

This property is available only when developing for Windows.

Gets an object that can be used to synchronize access to the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private Object System.Collections.ICollection.SyncRoot { get; }
```

Property Value

An object that can be used to synchronize access to the collection.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.System.Collections.IList.IsFixedSize Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether this list has a fixed size.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.IList.IsFixedSize { get; }
```

Property Value

true if the list has a fixed size; **false** otherwise.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.System.Collections.IList.IsReadOnly Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether this object is read-only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.IList.IsReadOnly { get; }
```

Property Value

true if this object is read-only; **false** otherwise.

See Also

Reference

[VertexChannel Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel Generic Class

Note

This generic class is available only when developing for Windows.

Provides methods and properties for maintaining a vertex channel.

This is a generic implementation of [VertexChannel](#) and, therefore, can handle strongly typed content data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class VertexChannel<T> : VertexChannel, IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

See Also

Reference

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)





[VertexChannelNames](#)

PlatformsWindows XP SP2, Windows Vista











VertexChannel Members

The following tables list the members exposed by the VertexChannel type.



Public Properties

Name	Description
 Count	(Inherited from VertexChannel .)
 ElementType	Gets the type of data contained in this channel.
 Item	Gets or sets the element at the specified index.
 Name	(Inherited from VertexChannel .)







Public Methods

Name	Description
 Contains	Overloaded. Determines whether the specified element is in the channel.
 CopyTo	Overloaded. Copies the elements of the channel to an array.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator interface for reading channel content.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Overloaded. Gets the index of the specified item.
 ReadConvertedContent	Reads channel content and automatically converts it to the specified vector format.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<T>.IsReadOnly	Gets a value indicating whether this object is read-only.
 System.Collections.Generic.ICollection<T>.Add	Adds a new element to the end of the collection.
 System.Collections.Generic.ICollection<T>.Clear	Removes all elements from the collection.
 System.Collections.Generic.IList<T>.Insert	Inserts an element into the collection at the specified position.
 System.Collections.Generic.IList<T>.RemoveAt	Removes the element at the specified index position.
 System.Collections.Generic.ICollection<T>.Remove	Removes a specified element from the collection.

See Also











Reference

[VertexChannel Generic Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel Methods







Public Methods

Name	Description
 Contains	Overloaded. Determines whether the specified element is in the channel.
 CopyTo	Overloaded. Copies the elements of the channel to an array.
 Equals	(Inherited from Object .)
 GetEnumerator	Gets an enumerator interface for reading channel content.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 IndexOf	Overloaded. Gets the index of the specified item.
 ReadConvertedContent	Reads channel content and automatically converts it to the specified vector format.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
 System.Collections.Generic.ICollection<T>.IsReadOnly	Gets a value indicating whether this object is read-only.
 System.Collections.Generic.ICollection<T>.Add	Adds a new element to the end of the collection.
 System.Collections.Generic.ICollection<T>.Clear	Removes all elements from the collection.
 System.Collections.Generic.IList<T>.Insert	Inserts an element into the collection at the specified position.
 System.Collections.Generic.IList<T>.RemoveAt	Removes the element at the specified index position.
 System.Collections.Generic.ICollection<T>.Remove	Removes a specified element from the collection.

See Also

Reference

[VertexChannel Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.Contains Method

Determines whether the specified element is in the channel.

Overload List

Name	Description
VertexChannel.Contains (T)	Determines whether the specified element is in the channel.
VertexChannel.Contains (Object)	(Inherited from VertexChannel .)

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.Contains Method (T)

Note

This method is available only when developing for Windows.

Determines whether the specified element is in the channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Contains (  
    T item  
)
```

Parameters

item

Element being searched for.

Return Value

true if the element is present; **false** otherwise.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.CopyTo Method

Copies the elements of the channel to an array.

Overload List

Name	Description
VertexChannel.CopyTo (T[], Int32)	Copies the elements of the channel to an array, starting at the specified index.
VertexChannel.CopyTo (Array, Int32)	(Inherited from VertexChannel .)

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.CopyTo Method (T[], Int32)

Note

This method is available only when developing for Windows.

Copies the elements of the channel to an array, starting at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void CopyTo (  
    T[] array,  
    int arrayIndex  
)
```

Parameters

array

Array that will receive the copied channel elements.

arrayIndex

Starting index for copy operation.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel.GetEnumerator Method

Note

This method is available only when developing for Windows.

Gets an enumerator interface for reading channel content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerable<T> GetEnumerator ()
```

Return Value

Enumeration of the channel content.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.IndexOf Method

Gets the index of the specified item.

Overload List

Name	Description
VertexChannel.IndexOf (T)	Gets the index of the specified item.
VertexChannel.IndexOf (Object)	(Inherited from VertexChannel .)

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.IndexOf Method (T)

Note

This method is available only when developing for Windows.

Gets the index of the specified item.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int IndexOf (  
    T item  
)
```

Parameters

item

Item whose index is to be retrieved.

Return Value

Index of specified item.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.ReadConvertedContent Generic Method

Note

This generic method is available only when developing for Windows.

Reads channel content and automatically converts it to the specified vector format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override IEnumerable<TargetType> ReadConvertedContent<TargetType> ()
```

Type Parameters

TargetType

Return Value

Enumeration of retrieved channel data.

Remarks

The following formats are supported:

- [Single](#)
- [Vector2 Structure](#)
- [Vector3 Structure](#)
- [Vector4 Structure](#)
- Any implementation of [IPackedVector Interface](#).

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<T>.Add Method

Note

This method is available only when developing for Windows.

Adds a new element to the end of the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<T>.Add (  
    T item  
)
```

Parameters

item

The element to add.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.ICollection<T>.Clear Method

Note

This method is available only when developing for Windows.

Removes all elements from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<T>.Clear ()
```

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<T>.Remove Method

Note

This method is available only when developing for Windows.

Removes a specified element from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<T>.Remove (  
    T item  
)
```

Parameters

item

The element to remove.

Return Value

true if the channel was removed; **false** otherwise.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.IList<T>.Insert Method

Note

This method is available only when developing for Windows.

Inserts an element into the collection at the specified position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<T>.Insert (  
    int index,  
    T item  
)
```

Parameters

index

Index at which to insert the element.

item

The element to insert.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.IList<T>.RemoveAt Method

Note

This method is available only when developing for Windows.

Removes the element at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<T>.RemoveAt (  
    int index  
)
```

Parameters

index

Index of the element to remove.

See Also

Reference

[VertexChannel Generic Class](#)





[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannel Properties

Public Properties

	Name	Description
	Count	(Inherited from VertexChannel .)
	ElementType	Gets the type of data contained in this channel.
	Item	Gets or sets the element at the specified index.
	Name	(Inherited from VertexChannel .)

See Also

Reference

[VertexChannel Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannel.ElementType Property

Note

This property is available only when developing for Windows.

Gets the type of data contained in this channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override Type ElementType { get; }
```

Property Value

Type of data stored in the vertex channel.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.Item Property

Note

This property is available only when developing for Windows.

Gets or sets the element at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T this [
    int index
] { get; set; }
```

Property Value

Element at the specified index.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannel.System.Collections.Generic.ICollection<T>.IsReadOnly Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether this object is read-only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<T>.IsReadOnly { get; }
```

Property Value

true if this object is read-only; **false** otherwise.

See Also

Reference

[VertexChannel Generic Class](#)

[VertexChannel Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing a list of vertex data channels.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class VertexChannelCollection : IList<VertexChannel>, ICollection<VertexChannel>, IEnumerable<VertexChannel>, IEnumerable
```

See Also

Reference

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)



[VertexChannelNames](#)

Platforms Windows XP SP2, Windows Vista
















VertexChannelCollection Members

The following tables list the members exposed by the VertexChannelCollection type.



Public Properties

	Name	Description
	Count	Gets the number of vertex channels in the collection.
	Item	Overloaded. Gets or sets a vertex channel from the collection.






Public Methods

	Name	Description
	Add	Overloaded. Adds a new vertex channel to the collection.
	Clear	Removes all vertex channels from the collection.
	Contains	Overloaded. Determines whether the collection contains the specified vertex channel.
	ConvertChannelContent	Overloaded. Converts a channel to another vector format.
	Equals	(Inherited from Object .)
	Get	Overloaded. Gets the vertex channel that matches the specified traits.
	GetEnumerator	Gets an enumerator that iterates through the vertex channels of a collection.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	IndexOf	Overloaded. Determines the index of a vertex channel.
	Insert	Overloaded. Inserts a new vertex channel into the collection.
	ReferenceEquals	(Inherited from Object .)
	Remove	Overloaded. Removes a specified vertex channel from the collection.
	RemoveAt	Removes the vertex channel at the specified index position.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

	Name	Description
	System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.IsReadOnly	Determines whether the collection is read-only.
	System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.Add	Adds a new vertex channel to the collection.
	System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.CopyTo	Copies the elements of the collection to an array, starting at the specified index.
	System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through a collection.
	System.Collections.Generic.IList<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.Insert	Inserts an item at the specified index.

See Also

Reference

[VertexChannelCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection Methods

Public Methods

Name	Description
Add	Overloaded. Adds a new vertex channel to the collection.
Clear	Removes all vertex channels from the collection.
Contains	Overloaded. Determines whether the collection contains the specified vertex channel.
ConvertChannelContent	Overloaded. Converts a channel to another vector format.
Equals	(Inherited from Object .)
Get	Overloaded. Gets the vertex channel that matches the specified traits.
GetEnumerator	Gets an enumerator that iterates through the vertex channels of a collection.
GetHashCode	(Inherited from Object .)
GetType	(Inherited from Object .)
IndexOf	Overloaded. Determines the index of a vertex channel.
Insert	Overloaded. Inserts a new vertex channel into the collection.
ReferenceEquals	(Inherited from Object .)
Remove	Overloaded. Removes a specified vertex channel from the collection.
RemoveAt	Removes the vertex channel at the specified index position.
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
MemberwiseClone	(Inherited from Object .)

Explicit Interface Implementations

Name	Description
System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.IsReadOnly	Determines whether the collection is read-only.
System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.Add	Adds a new vertex channel to the collection.
System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.CopyTo	Copies the elements of the collection to an array, starting at the specified index.
System.Collections.IEnumerable.GetEnumerator	Returns an enumerator that iterates through a collection.
System.Collections.Generic.IList<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.Insert	Inserts an item at the specified index.

See Also

Reference

[VertexChannelCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Add Method

Adds a new vertex channel to the collection.

Overload List

Name	Description
VertexChannelCollection.Add (String, Generic IEnumerable)	Adds a new vertex channel to the end of the collection.
VertexChannelCollection.Add (String, Type, IEnumerable)	Adds a new vertex channel to the end of the collection.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Add Generic Method (String, Generic IEnumerable)

Note

This generic method is available only when developing for Windows.

Adds a new vertex channel to the end of the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<ElementType> Add<ElementType> (  
    string name,  
    IEnumerable<ElementType> channelData  
)
```

Type Parameters

ElementType

Type of the channel.

Parameters

name

Name of the new channel.

channelData

Initial data for the new channel. If null, the channel is filled with the default value for that type.

Return Value

The newly added vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Add Method (String, Type, IEnumerable)

Note

This method is available only when developing for Windows.

Adds a new vertex channel to the end of the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel Add (  
    string name,  
    Type elementType,  
    IEnumerable channelData  
)
```

Parameters

name

Name of the new channel.

elementType

Type of data to be contained in the new channel.

channelData

Initial data for the new channel. If null, the channel is filled with the default value for that type.

Return Value

The newly added vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelCollection.Clear Method

Note

This method is available only when developing for Windows.

Removes all vertex channels from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Clear ()
```

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Contains Method

Determines whether the collection contains the specified vertex channel.

Overload List

Name	Description
VertexChannelCollection.Contains (String)	Determines whether the collection contains the specified vertex channel.
VertexChannelCollection.Contains (VertexChannel)	Determines whether the collection contains the specified vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Contains Method (String)

Note

This method is available only when developing for Windows.

Determines whether the collection contains the specified vertex channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Contains (  
    string name  
)
```

Parameters

name

Name of the channel being searched for.

Return Value

true if the channel was found; **false** otherwise.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Contains Method (VertexChannel)

Note

This method is available only when developing for Windows.

Determines whether the collection contains the specified vertex channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Contains (  
    VertexChannel item  
)
```

Parameters

item

Name of the channel being searched for.

Return Value

true if the channel was found; **false** otherwise.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelCollection.ConvertChannelContent Method

Converts a channel to another vector format.

Overload List

Name	Description
VertexChannelCollection.ConvertChannelContent (Int32)	Converts the channel, at the specified index, to another vector format.
VertexChannelCollection.ConvertChannelContent (String)	Converts the channel, specified by <i>name</i> to another vector format.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.ConvertChannelContent Generic Method (Int32)

Note

This generic method is available only when developing for Windows.

Converts the channel, at the specified index, to another vector format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<TargetType> ConvertChannelContent<TargetType> (  
    int index  
)
```

Type Parameters

TargetType

Type of the target format. Can be one of the following:

- [Single](#)
- [Vector2](#)
- [Vector3](#)
- [Vector4](#)
- [IPackedVector](#)

Parameters

index

Index of the channel to be converted.

Return Value

New channel in the specified format.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.ConvertChannelContent Generic Method (String)

Note

This generic method is available only when developing for Windows.

Converts the channel, specified by *name* to another vector format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<TargetType> ConvertChannelContent<TargetType> (  
    string name  
)
```

Type Parameters

TargetType

Type of the target format. Can be one of the following:

- [Single](#)
- [Vector2](#)
- [Vector3](#)
- [Vector4](#)
- [IPackedVector](#)

Parameters

name

Name of the channel to be converted.

Return Value

New channel in the specified format.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Get Method

Gets the vertex channel that matches the specified traits.

Overload List

Name	Description
VertexChannelCollection.Get (Int32)	Gets the vertex channel with the specified index and content type.
VertexChannelCollection.Get (String)	Gets the vertex channel with the specified name and content type.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Get Generic Method (Int32)

Note

This generic method is available only when developing for Windows.

Gets the vertex channel with the specified index and content type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<T> Get<T> (  
    int index  
)
```

Type Parameters

T

Parameters

index

Index of a vertex channel.

Return Value

Type of a vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Get Generic Method (String)

Note

This generic method is available only when developing for Windows.

Gets the vertex channel with the specified name and content type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<T> Get<T> (  
    string name  
)
```

Type Parameters

T

Type of the content.

Parameters

name

Name of a vertex channel.

Return Value

Type of a vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.GetEnumerator Method

Note

This method is available only when developing for Windows.

Gets an enumerator that iterates through the vertex channels of a collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IEnumerator<VertexChannel> GetEnumerator ()
```

Return Value

Enumerator for the collection.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.IndexOf Method

Determines the index of a vertex channel.

Overload List

Name	Description
VertexChannelCollection.IndexOf (String)	Determines the index of a vertex channel with the specified name.
VertexChannelCollection.IndexOf (VertexChannel)	Determines the index of the specified vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.IndexOf Method (String)

Note

This method is available only when developing for Windows.

Determines the index of a vertex channel with the specified name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int IndexOf (
    string name
)
```

Parameters

name

Name of the vertex channel being searched for.

Return Value

Index of the vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelCollection.IndexOf Method (VertexChannel)

Note

This method is available only when developing for Windows.

Determines the index of the specified vertex channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int IndexOf (
    VertexChannel item
)
```

Parameters

item

Vertex channel being searched for.

Return Value

Index of the vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelCollection.Insert Method

Inserts a new vertex channel into the collection.

Overload List

Name	Description
VertexChannelCollection.Insert (Int32, String, Generic IEnumerable)	Inserts a new vertex channel at the specified position.
VertexChannelCollection.Insert (Int32, String, Type, IEnumerable)	Inserts a new vertex channel at the specified position.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Insert Generic Method (Int32, String, Generic IEnumerable)

Note

This generic method is available only when developing for Windows.

Inserts a new vertex channel at the specified position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<ElementType> Insert<ElementType> (  
    int index,  
    string name,  
    IEnumerable<ElementType> channelData  
)
```

Type Parameters

ElementType

Parameters

index

Index for channel insertion.

name

Name of the new channel.

channelData

Type of the new channel.

Return Value

The inserted vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Insert Method (Int32, String, Type, IEnumerable)

Note

This method is available only when developing for Windows.

Inserts a new vertex channel at the specified position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel Insert (  
    int index,  
    string name,  
    Type elementType,  
    IEnumerable channelData  
)
```

Parameters

index

Index for channel insertion.

name

Name of the new channel.

elementType

Type of the new channel.

channelData

Initial data for the new channel. If null, it is filled with the default value.

Return Value

The inserted vertex channel.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Remove Method

Removes a specified vertex channel from the collection.

Overload List

Name	Description
VertexChannelCollection.Remove (String)	Removes the specified vertex channel from the collection.
VertexChannelCollection.Remove (VertexChannel)	Removes the specified vertex channel from the collection.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Remove Method (String)

Note

This method is available only when developing for Windows.

Removes the specified vertex channel from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Remove (  
    string name  
)
```

Parameters

name

Name of the vertex channel being removed.

Return Value

true if the channel was removed; **false** otherwise.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Remove Method (VertexChannel)

Note

This method is available only when developing for Windows.

Removes the specified vertex channel from the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool Remove (  
    VertexChannel item  
)
```

Parameters

item

Name of the vertex channel being removed.

Return Value

true if the channel was removed; **false** otherwise.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.RemoveAt Method

Note

This method is available only when developing for Windows.

Removes the vertex channel at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void RemoveAt (  
    int index  
)
```

Parameters

index

Index of the vertex channel being removed.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel> Add Method

Note

This method is available only when developing for Windows.

Adds a new vertex channel to the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.Add (
    VertexChannel item
)
```

Parameters

item

Vertex channel to be added.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexCh Method

Note

This method is available only when developing for Windows.

Copies the elements of the collection to an array, starting at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.CopyTo (
    VertexChannel[] array,
    int arrayIndex
)
```

Parameters

array

The destination array.

arrayIndex

The index at which to begin copying elements.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.Generic.IList<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel> Method

Note

This method is available only when developing for Windows.

Inserts an item at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private void System.Collections.Generic.IList<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.Insert (
    int index,
    VertexChannel item
)
```

Parameters

index

The zero-based index at which *item* should be inserted.

item

The item to insert.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

System.Collections.IEnumerable.GetEnumerator Method

Note

This method is available only when developing for Windows.

Returns an enumerator that iterates through a collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator ()
```

Return Value

An object that can be used to iterate through the collection.

See Also

Reference

[VertexChannelCollection Class](#)



[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection Properties

Public Properties

	Name	Description
	Count	Gets the number of vertex channels in the collection.
	Item	Overloaded. Gets or sets a vertex channel from the collection.

See Also

Reference

[VertexChannelCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Count Property

Note

This property is available only when developing for Windows.

Gets the number of vertex channels in the collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Count { get; }
```

Property Value

Number of vertex channels.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Item Property

Gets or sets a vertex channel from the collection.

Overload List

Name	Description
VertexChannelCollection.Item (Int32)	Gets or sets the vertex channel at the specified index position.
VertexChannelCollection.Item (String)	Gets or sets the vertex channel with the specified name.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelCollection.Item Property (Int32)

Note

This property is available only when developing for Windows.

Gets or sets the vertex channel at the specified index position.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel this [
    int index
] { get; set; }
```

Property Value

A vertex channel object.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelCollection.Item Property (String)

Note

This property is available only when developing for Windows.

Gets or sets the vertex channel with the specified name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel this [
    string name
] { get; }
```

Property Value

A vertex channel object.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelCollection.System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pi Property

Note

This property is available only when developing for Windows.

Determines whether the collection is read-only.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
private bool System.Collections.Generic.ICollection<Microsoft.Xna.Framework.Content.Pipeline.Graphics.VertexChannel>.IsReadOnly { get; }
```

Property Value

true if the collection is read-only; **false** otherwise.

See Also

Reference

[VertexChannelCollection Class](#)

[VertexChannelCollection Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames Class

Note

This class is available only when developing for Windows.

Provides properties for managing a collection of vertex channel names.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static class VertexChannelNames
```

Remarks

These vertex channel names can be passed to [VertexChannelCollection.Add](#) and looked up using the [Name](#) property of a specific vertex channel. The names are stored in the [VertexChannel.Name](#) property.

See Also

Reference

[VertexChannelNames Members](#)
















[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista



VertexChannelNames Members

The following tables list the members exposed by the VertexChannelNames type.

Public Methods

	Name	Description
	Binormal	Gets the name of a binormal vector channel with the specified index.
	Color	Gets the name of a color channel with the specified index.
	DecodeBaseName	Gets a channel base name stub from the encoded string format.
	DecodeUsageIndex	Gets a channel usage index from the encoded format.
	EncodeName	Overloaded. Combines a vertex string and usage index into an encoded string.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Normal	Overloaded. Gets the name of a normal channel.
	ReferenceEquals	(Inherited from Object .)
	Tangent	Gets the name of the tangent vector channel with the specified index.
	TextureCoordinate	Gets the name of the texture coordinate channel with the specified index.
	ToString	(Inherited from Object .)
	TryDecodeUsage	Gets a vertex declaration usage enumeration from the encoded string format.
	Weights	Overloaded. Gets the name of an animation weights channel.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


























Reference

[VertexChannelNames Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelNames Methods

Public Methods

	Name	Description
 	Binormal	Gets the name of a binormal vector channel with the specified index.
 	Color	Gets the name of a color channel with the specified index.
 	DecodeBaseName	Gets a channel base name stub from the encoded string format.
 	DecodeUsageIndex	Gets a channel usage index from the encoded format.
 	EncodeName	Overloaded. Combines a vertex string and usage index into an encoded string.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
 	Normal	Overloaded. Gets the name of a normal channel.
	ReferenceEquals	(Inherited from Object .)
 	Tangent	Gets the name of the tangent vector channel with the specified index.
 	TextureCoordinate	Gets the name of the texture coordinate channel with the specified index.
	ToString	(Inherited from Object .)
 	TryDecodeUsage	Gets a vertex declaration usage enumeration from the encoded string format.
 	Weights	Overloaded. Gets the name of an animation weights channel.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexChannelNames Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelNames.Binormal Method

Note

This method is available only when developing for Windows.

Gets the name of a binormal vector channel with the specified index.

This will typically contain [Vector3](#) data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Binormal (  
    int usageIndex  
)
```

Parameters

usageIndex

Zero-based index of the vector channel being retrieved.

Return Value

Name of the retrieved vector channel.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.Color Method

Note

This method is available only when developing for Windows.

Gets the name of a color channel with the specified index.

This will typically contain [Vector3](#) data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Color (  
    int usageIndex  
)
```

Parameters

usageIndex

Zero-based index of the color channel being retrieved.

Return Value

Name of the retrieved color channel.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.DecodeBaseName Method

Note

This method is available only when developing for Windows.

Gets a channel base name stub from the encoded string format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string DecodeBaseName (  
    string encodedName  
)
```

Parameters

encodedName

Encoded string to be decoded.

Return Value

Extracted base name.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.DecodeUsageIndex Method

Note

This method is available only when developing for Windows.

Gets a channel usage index from the encoded format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static int DecodeUsageIndex (  
    string encodedName  
)
```

Parameters

encodedName

Encoded name to be decoded.

Return Value

Resulting channel usage index.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.EncodeName Method

Combines a vertex string and usage index into an encoded string.

Overload List

Name	Description
VertexChannelNames.EncodeName (String, Int32)	Combines a channel name stub and usage index into a string name.
VertexChannelNames.EncodeName (VertexElementUsage, Int32)	Combines a vertex declaration usage and usage index into a string name.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelNames.EncodeName Method (String, Int32)

Note

This method is available only when developing for Windows.

Combines a channel name stub and usage index into a string name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string EncodeName (  
    string baseName,  
    int usageIndex  
)
```

Parameters

baseName

A channel base name stub.

usageIndex

A channel usage index.

Return Value

Resulting encoded name.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelNames.EncodeName Method

(VertexElementUsage, Int32)

Note

This method is available only when developing for Windows.

Combines a vertex declaration usage and usage index into a string name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string EncodeName (  
    VertexElementUsage vertexElementUsage,  
    int usageIndex  
)
```

Parameters

vertexElementUsage

A vertex declaration.

usageIndex

An index for the vertex declaration.

Return Value

Resulting encoded name.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.Normal Method

Gets the name of a normal channel.

This will typically contain [Vector3](#) data.

Overload List

Name	Description
VertexChannelNames.Normal ()	Gets the name of the primary normal channel.
VertexChannelNames.Normal (Int32)	Gets the name of the normal channel with the specified index.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelNames.Normal Method ()

Note

This method is available only when developing for Windows.

Gets the name of the primary normal channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Normal ()
```

Return Value

Primary normal channel name.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.Normal Method (Int32)

Note

This method is available only when developing for Windows.

Gets the name of the normal channel with the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Normal (  
    int usageIndex  
)
```

Parameters

usageIndex

Zero-based index of the normal channel being retrieved.

Return Value

Normal channel at the specified index.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexChannelNames.Tangent Method

Note

This method is available only when developing for Windows.

Gets the name of the tangent vector channel with the specified index.

This will typically contain [Vector3](#) data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Tangent (  
    int usageIndex  
)
```

Parameters

usageIndex

Zero-based index of the tangent vector channel being retrieved.

Return Value

Name of the retrieved tangent vector channel.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.TextureCoordinate Method

Note

This method is available only when developing for Windows.

Gets the name of the texture coordinate channel with the specified index.

This will typically contain [Vector3](#) data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string TextureCoordinate (  
    int usageIndex  
)
```

Parameters

usageIndex

Zero-based index of the texture coordinate channel being retrieved.

Return Value

Name of the retrieved texture coordinate channel.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.TryDecodeUsage Method

Note

This method is available only when developing for Windows.

Gets a vertex declaration usage enumeration from the encoded string format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static bool TryDecodeUsage (  
    string encodedName,  
    out VertexElementUsage usage  
)
```

Parameters

encodedName

Encoded name of a vertex declaration.

usage

[[OutAttribute](#)] Value of the declaration usage for the vertex declaration.

Return Value

true if the encoded name maps to a [VertexElementUsage](#) enumeration value; **false** otherwise.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.Weights Method

Gets the name of an animation weights channel.

This will typically contain data on the bone weights for a vertex channel. For more information, see [BoneWeightCollection](#).

Overload List

Name	Description
VertexChannelNames.Weights ()	Gets the name of the primary animation weights channel.
VertexChannelNames.Weights (Int32)	Gets the name of an animation weights channel at the specified index.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexChannelNames.Weights Method ()

Note

This method is available only when developing for Windows.

Gets the name of the primary animation weights channel.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Weights ()
```

Return Value

Name of the primary animation weights channel.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexChannelNames.Weights Method (Int32)

Note

This method is available only when developing for Windows.

Gets the name of an animation weights channel at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static string Weights (  
    int usageIndex  
)
```

Parameters

usageIndex

Index of the animation weight channel to be retrieved.

Return Value

Name of the retrieved animation weights channel.

See Also

Reference

[VertexChannelNames Class](#)

[VertexChannelNames Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for maintaining the vertex data of a [GeometryContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class VertexContent
```

Remarks

This class combines a collection of arbitrarily named data channels with a list of position indices that reference the [Positions](#) collection of the parent [MeshContent](#).

See Also

Reference

[VertexContent Members](#)





[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista













VertexContent Members

The following tables list the members exposed by the VertexContent type.



Public Properties

Name	Description
 Channels	Gets the list of named vertex data channels in the VertexContent .
 PositionIndices	Gets the list of position indices.
 Positions	Gets position data from the parent mesh object.
 VertexCount	Number of vertices for the content.

Public Methods

Name	Description
 Add	Appends a new vertex index to the end of the PositionIndices collection.
 AddRange	Appends multiple vertex indices to the end of the PositionIndices collection.
 CreateVertexBuffer	Converts design-time vertex position and channel data into a vertex buffer format that a graphics device can recognize.
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Insert	Inserts a new vertex index to the PositionIndices collection.
 InsertRange	Inserts multiple vertex indices to the PositionIndices collection.
 ReferenceEquals	(Inherited from Object .)
 RemoveAt	Removes a vertex index from the specified location in both PositionIndices and VertexChannel<T> .
 RemoveRange	Removes a range of vertex indices from the specified location in both PositionIndices and VertexChannel<T> .
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexContent Methods

Public Methods

Name	Description
Add	Appends a new vertex index to the end of the PositionIndices collection.
AddRange	Appends multiple vertex indices to the end of the PositionIndices collection.
CreateVertexBuffer	Converts design-time vertex position and channel data into a vertex buffer format that a graphics device can recognize.
Equals	(Inherited from Object .)
GetHashCode	(Inherited from Object .)
GetType	(Inherited from Object .)
Insert	Inserts a new vertex index to the PositionIndices collection.
InsertRange	Inserts multiple vertex indices to the PositionIndices collection.
ReferenceEquals	(Inherited from Object .)
RemoveAt	Removes a vertex index from the specified location in both PositionIndices and VertexChannel<T> .
RemoveRange	Removes a range of vertex indices from the specified location in both PositionIndices and VertexChannel<T> .
ToString	(Inherited from Object .)

Protected Methods

Name	Description
Finalize	(Inherited from Object .)
MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexContent.Add Method

Note

This method is available only when developing for Windows.

Appends a new vertex index to the end of the [PositionIndices](#) collection.

Other vertex channels will automatically be extended and the new indices populated with default values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Add (  
    int positionIndex  
)
```

Parameters

positionIndex

Index into the [MeshContent.Positions](#) member of the parent.

Return Value

Index of the new entry. This can be added to the [Indices](#) member of the parent.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.AddRange Method

Note

This method is available only when developing for Windows.

Appends multiple vertex indices to the end of the [PositionIndices](#) collection.

Other vertex channels will automatically be extended and the new indices populated with default values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void AddRange (  
    IEnumerable<int> positionIndexCollection  
)
```

Parameters

positionIndexCollection

Index into the [Positions](#) member of the parent.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.CreateVertexBuffer Method

Note

This method is available only when developing for Windows.

Converts design-time vertex position and channel data into a vertex buffer format that a graphics device can recognize.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void CreateVertexBuffer (  
    out VertexBufferContent vertexBuffer,  
    out VertexElement[] vertexElements,  
    TargetPlatform targetPlatform  
)
```

Parameters

vertexBuffer

[[OutAttribute](#)] A packed vertex buffer.

vertexElements

[[OutAttribute](#)] Description of the vertex format.

targetPlatform

Target platform of the content build.

Remarks

This method returns both a packed vertex buffer and a description of the vertex format.

Exceptions

Exception type	Condition
InvalidContentException	One or more of the vertex channel types are invalid or an unrecognized name was passed to VertexElementUsage .

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.Insert Method

Note

This method is available only when developing for Windows.

Inserts a new vertex index to the [PositionIndices](#) collection.

Other vertex channels will automatically be extended and the new indices populated with default values.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Insert (  
    int index,  
    int positionIndex  
)
```

Parameters

index

Vertex index to be inserted.

positionIndex

Position of new vertex index in the collection.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.InsertRange Method

Note

This method is available only when developing for Windows.

Inserts multiple vertex indices to the [PositionIndices](#) collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void InsertRange (  
    int index,  
    IEnumerable<int> positionIndexCollection  
)
```

Parameters

index

Vertex index to be inserted.

positionIndexCollection

Position of the first element of the inserted range in the collection.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexContent.RemoveAt Method

Note

This method is available only when developing for Windows.

Removes a vertex index from the specified location in both [PositionIndices](#) and [VertexChannel<T>](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void RemoveAt (  
    int index  
)
```

Parameters

index

Index of the vertex to be removed.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.RemoveRange Method

Note

This method is available only when developing for Windows.

Removes a range of vertex indices from the specified location in both [PositionIndices](#) and [VertexChannel<T>](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void RemoveRange (  
    int index,  
    int count  
)
```

Parameters

index

Index of the first vertex index to be removed.

count

Number of indices to remove.

See Also

Reference

[VertexContent Class](#)





[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

Platforms Windows XP SP2, Windows Vista

VertexContent Properties

Public Properties

	Name	Description
	Channels	Gets the list of named vertex data channels in the VertexContent .
	PositionIndices	Gets the list of position indices.
	Positions	Gets position data from the parent mesh object.
	VertexCount	Number of vertices for the content.

See Also

Reference

[VertexContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

VertexContent.Channels Property

Note

This property is available only when developing for Windows.

Gets the list of named vertex data channels in the [VertexContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannelCollection Channels { get; }
```

Property Value

Collection of vertex data channels.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.PositionIndices Property

Note

This property is available only when developing for Windows.

Gets the list of position indices.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexChannel<int> PositionIndices { get; }
```

Property Value

Position of the position index being retrieved.

Remarks

This list adds a level of indirection between the actual triangle indices and the [Positions](#) member of the parent. This indirection preserves the topological vertex identity in cases where a single vertex position is used by triangles that straddle a discontinuity in some other data channel.

For example, the following code gets the position of the first vertex of the first triangle in a [GeometryContent](#) object:

```
parent.Positions[Vertices.PositionIndices[Indices[0]]]
```

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.Positions Property

Note

This property is available only when developing for Windows.

Gets position data from the parent mesh object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IndirectPositionCollection Positions { get; }
```

Property Value

Collection of vertex positions for the mesh.

Remarks

The collection returned from this call provides a virtualized view of the vertex positions for this batch. The collection uses the contents of the [PositionIndices](#) property to index into the parent [Positions](#). This collection is read-only. If you need to modify any contained values, edit the [PositionIndices](#) or [Positions](#) members directly.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexContent.VertexCount Property

Note

This property is available only when developing for Windows.

Number of vertices for the content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Graphics

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int VertexCount { get; }
```

Property Value

Number of vertices.

See Also

Reference

[VertexContent Class](#)

[VertexContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Graphics Namespace](#)

PlatformsWindows XP SP2, Windows Vista

Microsoft.Xna.Framework.Content.Pipeline.Processors
























Namespace

Note


This namespace is available only when developing for Windows.

Provides base classes that represent processors used by the XNA Framework Content Pipeline when processing specific game asset types.

Classes

Name	Description
 EffectProcessor	Processes a string representation to a platform-specific compiled effect.
 FontDescriptionProcessor	Provides methods to convert a font description class containing the name and size of a font into SpriteFontContent .
 FontTextureProcessor	Provides methods to convert a specially marked 2D bitmap into SpriteFontContent .
 MaterialProcessor	Builds the effects and textures of a MaterialContent object.
 ModelBoneContent	Stores design-time data for a ModelBone asset.
 ModelBoneContentCollection	A collection of ModelBoneContent objects.
 ModelContent	Stores design-time data for a Model asset.
 ModelMeshContent	Stores design-time data for a ModelMesh asset.
 ModelMeshContentCollection	A collection of ModelMeshContent objects.
 ModelMeshPartContent	Stores design-time data for a ModelMeshPart asset.
 ModelMeshPartContentCollection	Collection of ModelMeshPartContent objects.
 ModelProcessor	Processes a game asset mesh to a model content that is optimal for runtime.
 ModelTextureProcessor	Processes textures for 3D models.
 PassThroughProcessor	Provides an implementation of a no-operation processor that returns an unmodified copy of the input data.
 SongContent	Represents a processed Song object.
 SongProcessor	A custom song processor that processes an intermediate AudioContent type.
 SoundEffectContent	Represents a processed sound effect.
 SoundEffectProcessor	Provides a custom sound effect processor that processes an intermediate AudioContent type.
 SpriteFontContent	Provides methods and properties for managing a design-time SpriteFont Class holding packed font data.
 SpriteTextureProcessor	Processes textures for use as 2D sprites or user interface components.
 TextureProcessor	Provides methods for processing textures.
 VertexBufferContent	Provides methods and properties for managing a design-time vertex buffer that holds packed vertex data.
 VideoProcessor	Processes videos for playback within a game.

Enumerations

Name	Description
 TextureProcessorOutputFormat	Specifies the target output (of type SurfaceFormat) of the texture processor.

EffectProcessor Class

Note

This class is available only when developing for Windows.

Processes a string representation to a platform-specific compiled effect.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]  
public class EffectProcessor : ContentProcessor<EffectContent, CompiledEffect>
```

See Also

Reference

[EffectProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista







EffectProcessor Members

The following tables list the members exposed by the EffectProcessor type.



Public Constructors

Name	Description
 EffectProcessor	Initializes a new instance of EffectProcessor.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Processes the string representation of the specified effect into a platform-specific binary format using the specified context.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

EffectProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **EffectProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public EffectProcessor ()
```

See Also

Reference

[EffectProcessor Class](#)







[EffectProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

EffectProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Processes the string representation of the specified effect into a platform-specific binary format using the specified context.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[EffectProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

EffectProcessor.Process Method

Note

This method is available only when developing for Windows.

Processes the string representation of the specified effect into a platform-specific binary format using the specified context.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override CompiledEffect Process (
    EffectContent input,
    ContentProcessorContext context
)
```

Parameters

input

The effect string to be processed.

context

Context for the specified processor.

Return Value

A platform-specific compiled binary effect.

Remarks

If you get an error during processing, compilation stops immediately. The effect processor displays an error message. Once you fix the current error, it is possible you may get more errors on subsequent compilation attempts.

Note

When processing effects for the Xbox 360 platform, the following conditional compilation symbols are defined: XBOX and XBOX360. For more information, see [Cross-Platform Conditional Compilation Symbols](#).

See Also

Reference

[EffectProcessor Class](#)

[EffectProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontDescriptionProcessor Class

Note

This class is available only when developing for Windows.

Provides methods to convert a font description class containing the name and size of a font into [SpriteFontContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]
public class FontDescriptionProcessor : ContentProcessor<FontDescription, SpriteFontContent
>
```

See Also

Reference

[FontDescriptionProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Tasks


[How To: Extend the Font Description Processor to Support Additional Characters](#)

PlatformsWindows XP SP2, Windows Vista







FontDescriptionProcessor Members

The following tables list the members exposed by the FontDescriptionProcessor type.



Public Constructors

Name	Description
 FontDescriptionProcessor	Initializes a new instance of FontDescriptionProcessor.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Creates SpriteFontContent from a FontDescription object.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontDescriptionProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

FontDescriptionProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **FontDescriptionProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontDescriptionProcessor ()
```

See Also

Reference

[FontDescriptionProcessor Class](#)







[FontDescriptionProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

FontDescriptionProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Creates SpriteFontContent from a FontDescription object.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontDescriptionProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

FontDescriptionProcessor.Process Method

Note

This method is available only when developing for Windows.

Creates [SpriteFontContent](#) from a [FontDescription](#) object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override SpriteFontContent Process (  
    FontDescription input,  
    ContentProcessorContext context  
)
```

Parameters

input

Description of the font to build.

context

Context for the specified processor.

Return Value

The font data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>input</i> is null .

See Also

Reference

[FontDescriptionProcessor Class](#)

[FontDescriptionProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontTextureProcessor Class

Note

This class is available only when developing for Windows.

Provides methods to convert a specially marked 2D bitmap into [SpriteFontContent](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]
public class FontTextureProcessor : ContentProcessor<Texture2DContent, SpriteFontContent>
```

Remarks

The [FontTextureProcessor](#) class transforms a specially formatted bitmap-font texture into a [SpriteFontContent](#) object that can be loaded at run time as a standard [SpriteFont](#) object.

A bitmap-font texture must not contain more than one glyph for any given character code point; otherwise, it would not be clear which to use.

You can create a bitmap-font texture in the appropriate format using almost any raster drawing program. Place all the characters of the font in order on the bitmap as follows:

Each character should be arranged on the bitmap in a grid ordered from upper left to lower right. Monochrome characters should use white for solid areas and black for transparent areas. To include multicolored characters, add an alpha channel to the bitmap and use that to control which parts of the character are solid. Fill the spaces between each character and around the edges of the grid with a specific magenta color having the value Red:255 Green:0 Blue:255 Alpha:255. The processor packs the characters together as tightly as possible, eliminating any wasted space in the original grid.

The following figure shows an example of this kind of font texture.



The default behavior of [FontTextureProcessor](#) is to expect the first character at the upper left corner of the grid to be a space character (ASCII and Unicode code 32), and the one beside it to be an exclamation point (ASCII and Unicode code 33), and the succeeding characters to appear in Unicode order for all the characters that the font contains.

You can support a different range and order of characters by writing a custom processor that extends [FontTextureProcessor](#) by overriding the [GetCharacterForIndex](#) method. For example, the code below creates a processor class that you could use to load a bitmap font texture containing the numeric digits from 1 to 9:

C#

```
[ContentProcessor]
public class MyFontTextureProcessor : FontTextureProcessor
{
    protected override char GetCharacterForIndex( int index )
    {
        return (char)('1' + index);
    }
}
```

Be sure to include the `[ContentProcessor]` attribute before the class declaration, as shown in this example.

See Also

Reference

[FontTextureProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista



FontTextureProcessor Members

The following tables list the members exposed by the FontTextureProcessor type.







Public Constructors

Name	Description
 FontTextureProcessor	Initializes a new instance of FontTextureProcessor.




Public Properties

Name	Description
 FirstCharacter	Gets or sets the value of the First Character processor parameter.
 TextureFormat	Gets or sets the output SurfaceFormat of textures.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Creates SpriteFontContent from a bitmap.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 GetCharacterForIndex	Maps a glyph index to a character.
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

FontTextureProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **FontTextureProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public FontTextureProcessor ()
```

See Also

Reference

[FontTextureProcessor Class](#)







[FontTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)




Platforms Windows XP SP2, Windows Vista

FontTextureProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Creates SpriteFontContent from a bitmap.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	GetCharacterForIndex	Maps a glyph index to a character.
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[FontTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

FontTextureProcessor.GetCharacterForIndex Method

Note

This method is available only when developing for Windows.

Maps a glyph index to a character.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual char GetCharacterForIndex (
    int index
)
```

Parameters

index

The glyph index. This is a zero-based index that corresponds to the position of the glyph in the original character. This value begins at the upper-left corner of the texture and increases when moving to the right and down.

Return Value

The character represented by the specified glyph index.

Exceptions

Exception type	Condition
ArgumentOutOfRangeException	<i>index</i> is out of the range of available characters.

Remarks

This method is designed to be overridden by a subclass of this processor. For example, if the original texture contains the characters A, B, and C, in that order, this method should return "A" when passed 0, "B" when passed 1, and "C" when passed 2. Subclasses of this processor may override this method to control which characters a bitmap font represents.

If not overridden, the default implementation of this function assumes a Unicode order of character code points starting at code point 32—the space character. In other words, it assumes that a character index of zero corresponds to a space character, a character index of 1 corresponds to an exclamation point ("!"), and so on.

A font texture must not contain more than one instance of any given character code point. For example, it must not contain two uppercase "A" characters. As a result, any implementation of this function should ensure that it cannot return the same character for two different indices.

See Also

Reference

[FontTextureProcessor Class](#)

[FontTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontTextureProcessor.Process Method

Note

This method is available only when developing for Windows.

Creates [SpriteFontContent](#) from a bitmap.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override SpriteFontContent Process (
    Texture2DContent input,
    ContentProcessorContext context
)
```

Parameters

input

The bitmap data to convert to a font.

context

Context for the specified processor.

Return Value

The font data.

Exceptions

Exception type	Condition
ArgumentNullException	<i>input</i> is null .
InvalidOperationException	A character cannot be mapped to a glyph index: it is already mapped to a glyph index.

Remarks

Each character should be arranged on the bitmap in a grid ordered from upper left to lower right. Monochrome characters should use white for solid areas and black for transparent areas. To include multicolored characters, add an alpha channel to the bitmap and use that to control which parts of the character are solid. Fill the spaces between each character and around the edges of the grid with a specific magenta color having the value Red:255 Green:0 Blue:255 Alpha:255. The processor packs the characters together as tightly as possible, eliminating any wasted space in the original grid.

See Also

Reference

[FontTextureProcessor Class](#)



[FontTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontTextureProcessor Properties

Public Properties

	Name	Description
	FirstCharacter	Gets or sets the value of the First Character processor parameter.
	TextureFormat	Gets or sets the output SurfaceFormat of textures.

See Also

Reference

[FontTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

FontTextureProcessor.FirstCharacter Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the First Character processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual char FirstCharacter { get; set; }
```

Property Value

The start character of the font texture being processed.

See Also

Reference

[FontTextureProcessor Class](#)

[FontTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

FontTextureProcessor.TextureFormat Property

Note

This property is available only when developing for Windows.

Gets or sets the output [SurfaceFormat](#) of textures.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual TextureProcessorOutputFormat TextureFormat { get; set; }
```

Property Value

The output [SurfaceFormat](#) of textures.

Remarks

Textures can be left unchanged from the source asset, converted to the [Color](#) format, or DXT compressed.

See Also

Reference

[FontTextureProcessor Class](#)

[FontTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

MaterialProcessor Class

Note

This class is available only when developing for Windows.

Builds the effects and textures of a [MaterialContent](#) object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]  
public class MaterialProcessor : ContentProcessor<MaterialContent, MaterialContent>
```

See Also

Reference

[MaterialProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista






MaterialProcessor Members

The following tables list the members exposed by the MaterialProcessor type.







Public Constructors

Name	Description
 MaterialProcessor	Initializes a new instance of the MaterialProcessor class.





Public Properties

Name	Description
 ColorKeyColor	Gets or sets the color value to replace with transparent black.
 ColorKeyEnabled	Specifies whether color keying of a texture is enabled.
 GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source material. Existing mipmaps of the material are not replaced.
 ResizeTexturesToPowerOfTwo	Specifies whether resizing of a material is enabled.
 TextureFormat	Specifies the texture format of output materials.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Builds the texture and effect content for the material.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 BuildEffect	Builds effect content.
 BuildTexture	Builds texture content.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MaterialProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

MaterialProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [MaterialProcessor](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MaterialProcessor ()
```

See Also

Reference

[MaterialProcessor Class](#)







[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)





Platforms Windows XP SP2, Windows Vista

MaterialProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Builds the texture and effect content for the material.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	BuildEffect	Builds effect content.
	BuildTexture	Builds texture content.
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[MaterialProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

MaterialProcessor.BuildEffect Method

Note

This method is available only when developing for Windows.

Builds effect content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual ExternalReference<CompiledEffect> BuildEffect (  
    ExternalReference<EffectContent> effect,  
    ContentProcessorContext context  
)
```

Parameters

effect

An external reference to the effect content.

context

Context for the specified processor.

Return Value

A platform-specific compiled binary effect.

RemarksIf the input to process is of type [EffectMaterialContent](#), this function will be called to request that the [EffectContent](#) be built. The [EffectProcessor](#) is used to process the [EffectContent](#). Subclasses of [MaterialProcessor](#) can override this function to modify the parameters used to build [EffectContent](#). For example, a different version of this function could request a different processor for the [EffectContent](#).

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor.BuildTexture Method

Note

This method is available only when developing for Windows.

Builds texture content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual ExternalReference<TextureContent> BuildTexture (  
    string textureName,  
    ExternalReference<TextureContent> texture,  
    ContentProcessorContext context  
)
```

Parameters

textureName

The name of the texture. This should correspond to the key used to store the texture in [Textures](#).

texture

The asset to build. This should be a member of [Textures](#).

context

Context for the specified processor.

Return Value

The built texture content.

Remarks *textureName* can be used to determine which processor to use. For example, if a texture is being used as a normal map, the user may not want to use the [ModelTextureProcessor](#) on it, which compresses textures.

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor.Process Method

Note

This method is available only when developing for Windows.

Builds the texture and effect content for the material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override MaterialContent Process (  
    MaterialContent input,  
    ContentProcessorContext context  
)
```

Parameters

input

The material content to build.

context

Context for the specified processor.

Return Value

The built material.

RemarksIf the [MaterialContent](#) is of type [EffectMaterialContent](#), a build is requested for [Effect](#), and validation will be performed on the [OpaqueData](#) to ensure that all parameters are valid input to [SetValue](#) or [SetValueTranspose](#). If the [MaterialContent](#) is a [BasicMaterialContent](#), no validation will be performed on [OpaqueData](#). **Process** requests builds for all textures in [Textures](#), unless the [MaterialContent](#) is of type [BasicMaterialContent](#), in which case a build will only be requested for [DiffuseColor](#). The textures in [Textures](#) will be ignored.

See Also

Reference

[MaterialProcessor Class](#)






[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor Properties

Public Properties

	Name	Description
	ColorKeyColor	Gets or sets the color value to replace with transparent black.
	ColorKeyEnabled	Specifies whether color keying of a texture is enabled.
	GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source material. Existing mipmaps of the material are not replaced.
	ResizeTexturesToPowerOfTwo	Specifies whether resizing of a material is enabled.
	TextureFormat	Specifies the texture format of output materials.

See Also

Reference

[MaterialProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

MaterialProcessor.ColorKeyColor Property

Note

This property is available only when developing for Windows.

Gets or sets the color value to replace with transparent black.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual Color ColorKeyColor { get; set; }
```

Property Value

Color value of the material to replace with transparent black.

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor.ColorKeyEnabled Property

Note

This property is available only when developing for Windows.

Specifies whether color keying of a texture is enabled.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ColorKeyEnabled { get; set; }
```

Property Value

true if color keying is enabled; **false** otherwise.

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor.GenerateMipmaps Property

Note

This property is available only when developing for Windows.

Specifies if a full chain of mipmaps are generated from the source material. Existing mipmaps of the material are not replaced.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool GenerateMipmaps { get; set; }
```

Property Value

true if mipmap generation is enabled; **false** otherwise.

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor.ResizeTexturesToPowerOfTwo Property

Note

This property is available only when developing for Windows.

Specifies whether resizing of a material is enabled. Typically used to maximize compatibility with a graphics card because many graphics cards do not support a material size that is not a power of two. If **ResizeTexturesToPowerOfTwo** is enabled, the material is resized to the next largest power of two.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ResizeTexturesToPowerOfTwo { get; set; }
```

Property Value

true if resizing is enabled; **false** otherwise.

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

MaterialProcessor.TextureFormat Property

Note

This property is available only when developing for Windows.

Specifies the texture format of output materials. Materials can either be left unchanged from the source asset, converted to a corresponding [Color](#), or compressed using the appropriate **DXTCompressed** format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual TextureProcessorOutputFormat TextureFormat { get; set; }
```

Property Value

The texture format of the output.

See Also

Reference

[MaterialProcessor Class](#)

[MaterialProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelBoneContent Class

Note

This class is available only when developing for Windows.

Stores design-time data for a [ModelBone](#) asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelBoneContent
```

See Also

Reference

[ModelBoneContent Members](#)






[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista






ModelBoneContent Members

The following tables list the members exposed by the ModelBoneContent type.



Public Properties

	Name	Description
	Children	Gets the children of this bone.
	Index	Gets the index of this bone in a ModelBoneContentCollection .
	Name	Gets the bone name.
	Parent	Gets the parent of this bone.
	Transform	Gets or sets the local transformation matrix for this bone.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelBoneContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelBoneContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelBoneContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelBoneContent Properties

Public Properties

	Name	Description
	Children	Gets the children of this bone.
	Index	Gets the index of this bone in a ModelBoneContentCollection .
	Name	Gets the bone name.
	Parent	Gets the parent of this bone.
	Transform	Gets or sets the local transformation matrix for this bone.

See Also

Reference

[ModelBoneContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelBoneContent.Children Property

Note

This property is available only when developing for Windows.

Gets the children of this bone.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelBoneContentCollection Children { get; set; }
```

Property Value

The children of this bone.

See Also

Reference

[ModelBoneContent Class](#)

[ModelBoneContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelBoneContent.Index Property

Note

This property is available only when developing for Windows.

Gets the index of this bone in a [ModelBoneContentCollection](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int Index { get; }
```

Property Value

The index of this bone in a [ModelBoneContentCollection](#).

See Also

Reference

[ModelBoneContent Class](#)

[ModelBoneContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelBoneContent.Name Property

Note

This property is available only when developing for Windows.

Gets the bone name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The bone name.

See Also

Reference

[ModelBoneContent Class](#)

[ModelBoneContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelBoneContent.Parent Property

Note

This property is available only when developing for Windows.

Gets the parent of this bone.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelBoneContent Parent { get; }
```

Property Value

The parent of this bone.

See Also

Reference

[ModelBoneContent Class](#)

[ModelBoneContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelBoneContent.Transform Property

Note

This property is available only when developing for Windows.

Gets or sets the local transformation matrix for this bone.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Matrix Transform { get; set; }
```

Property Value

The local transformation matrix for this bone.

See Also

Reference

[ModelBoneContent Class](#)

[ModelBoneContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelBoneContentCollection Class

Note

This class is available only when developing for Windows.

A collection of [ModelBoneContent](#) objects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelBoneContentCollection : ReadOnlyCollection<ModelBoneContent>
```

See Also

Reference

[ModelBoneContentCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista


ModelBoneContentCollection Members

The following tables list the members exposed by the ModelBoneContentCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[ModelBoneContentCollection Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelBoneContentCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also



Reference

[ModelBoneContentCollection Class](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelBoneContentCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelBoneContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelContent Class

Note

This class is available only when developing for Windows.

Stores design-time data for a [Model](#) asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelContent
```

See Also

Reference

[ModelContent Members](#)





[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista






ModelContent Members

The following tables list the members exposed by the ModelContent type.



Public Properties

	Name	Description
	Bones	Gets the collection of bones that are referenced by this model.
	Meshes	Gets the collection of meshes that are associated with this model.
	Root	Gets the root bone of this model
	Tag	Gets a user defined tag object.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also





Reference

[ModelContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelContent Properties

Public Properties

	Name	Description
	Bones	Gets the collection of bones that are referenced by this model.
	Meshes	Gets the collection of meshes that are associated with this model.
	Root	Gets the root bone of this model
	Tag	Gets a user defined tag object.

See Also

Reference

[ModelContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelContent.Bones Property

Note

This property is available only when developing for Windows.

Gets the collection of bones that are referenced by this model.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelBoneContentCollection Bones { get; }
```

Property Value

The collection of bones that are referenced by this model.

See Also

Reference

[ModelContent Class](#)

[ModelContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelContent.Meshes Property

Note

This property is available only when developing for Windows.

Gets the collection of meshes that are associated with this model.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelMeshContentCollection Meshes { get; }
```

Property Value

The collection of meshes that are associated with this model.

See Also

Reference

[ModelContent Class](#)

[ModelContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelContent.Root Property

Note

This property is available only when developing for Windows.

Gets the root bone of this model

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelBoneContent Root { get; }
```

Property Value

The root bone of this model.

See Also

Reference

[ModelContent Class](#)

[ModelContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelContent.Tag Property

Note

This property is available only when developing for Windows.

Gets a user defined tag object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

User-defined tag object.

See Also

Reference

[ModelContent Class](#)

[ModelContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent Class

Note

This class is available only when developing for Windows.

Stores design-time data for a [ModelMesh](#) asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelMeshContent
```

See Also

Reference

[ModelMeshContent Members](#)









[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista






ModelMeshContent Members

The following tables list the members exposed by the ModelMeshContent type.



Public Properties

Name	Description
 BoundingSphere	Gets the bounding sphere for this mesh.
 IndexBuffer	Gets the index buffer associated with this mesh.
 MeshParts	Gets the children mesh parts associated with this mesh.
 Name	Gets the mesh name.
 ParentBone	Gets the parent bone.
 SourceMesh	Gets the MeshContent source object used when creating this class.
 Tag	Gets a user-defined tag object.
 VertexBuffer	Gets the vertex buffer associated with this mesh.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelMeshContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also









Reference

[ModelMeshContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshContent Properties

Public Properties

	Name	Description
	BoundingSphere	Gets the bounding sphere for this mesh.
	IndexBuffer	Gets the index buffer associated with this mesh.
	MeshParts	Gets the children mesh parts associated with this mesh.
	Name	Gets the mesh name.
	ParentBone	Gets the parent bone.
	SourceMesh	Gets the MeshContent source object used when creating this class.
	Tag	Gets a user-defined tag object.
	VertexBuffer	Gets the vertex buffer associated with this mesh.

See Also

Reference

[ModelMeshContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshContent.BoundingBox Property

Note

This property is available only when developing for Windows.

Gets the bounding sphere for this mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BoundingBox BoundingBox { get; }
```

Property Value

The bounding sphere for this mesh.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.IndexBuffer Property

Note

This property is available only when developing for Windows.

Gets the index buffer associated with this mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IndexCollection IndexBuffer { get; }
```

Property Value

The index buffer associated with this mesh.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.MeshParts Property

Note

This property is available only when developing for Windows.

Gets the children mesh parts associated with this mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelMeshPartContentCollection MeshParts { get; }
```

Property Value

The children mesh parts associated with this mesh.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.Name Property

Note

This property is available only when developing for Windows.

Gets the mesh name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string Name { get; }
```

Property Value

The mesh name.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.ParentBone Property

Note

This property is available only when developing for Windows.

Gets the parent bone.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelBoneContent ParentBone { get; }
```

Property Value

The parent bone.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.SourceMesh Property

Note

This property is available only when developing for Windows.

Gets the [MeshContent](#) source object used when creating this class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MeshContent SourceMesh { get; }
```

Property Value

The source mesh.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.Tag Property

Note

This property is available only when developing for Windows.

Gets a user-defined tag object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

User-defined tag object.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContent.VertexBuffer Property

Note

This property is available only when developing for Windows.

Gets the vertex buffer associated with this mesh.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexBufferContent VertexBuffer { get; }
```

Property Value

The vertex buffer associated with this mesh.

See Also

Reference

[ModelMeshContent Class](#)

[ModelMeshContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshContentCollection Class

Note

This class is available only when developing for Windows.

A collection of [ModelMeshContent](#) objects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelMeshContentCollection : ReadOnlyCollection<ModelMeshContent>
```

See Also

Reference

[ModelMeshContentCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista


ModelMeshContentCollection Members

The following tables list the members exposed by the ModelMeshContentCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[ModelMeshContentCollection Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshContentCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also



Reference

[ModelMeshContentCollection Class](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshContentCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelMeshContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshPartContent Class

Note

This class is available only when developing for Windows.

Stores design-time data for a [ModelMeshPart](#) asset.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelMeshPartContent
```

See Also

Reference

[ModelMeshPartContent Members](#)








[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista







ModelMeshPartContent Members

The following tables list the members exposed by the ModelMeshPartContent type.



Public Properties

Name	Description
 BaseVertex	Gets the offset from the start of the index buffer to the first vertex index.
 Material	Gets the material of this mesh part.
 NumVertices	Gets the number of vertices used in this mesh part.
 PrimitiveCount	Gets the number of primitives to render for this mesh part.
 StartIndex	Gets the location in the index buffer at which to start reading vertices.
 StreamOffset	Gets the offset, in bytes, from the first index of the of vertex buffer for this mesh part.
 Tag	Gets a user-defined tag object.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 GetVertexDeclaration	Gets the vertex declaration for this mesh part.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also







Reference

[ModelMeshPartContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshPartContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetVertexDeclaration	Gets the vertex declaration for this mesh part.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ModelMeshPartContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshPartContent.GetVertexDeclaration Method

Note

This method is available only when developing for Windows.

Gets the vertex declaration for this mesh part.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexElement[] GetVertexDeclaration ()
```

Return Value

The vertex declaration for this mesh part.

See Also

Reference

[ModelMeshPartContent Class](#)








[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent Properties

Public Properties

	Name	Description
	BaseVertex	Gets the offset from the start of the index buffer to the first vertex index.
	Material	Gets the material of this mesh part.
	NumVertices	Gets the number of vertices used in this mesh part.
	PrimitiveCount	Gets the number of primitives to render for this mesh part.
	StartIndex	Gets the location in the index buffer at which to start reading vertices.
	StreamOffset	Gets the offset, in bytes, from the first index of the of vertex buffer for this mesh part.
	Tag	Gets a user-defined tag object.

See Also

Reference

[ModelMeshPartContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshPartContent.BaseVertex Property

Note

This property is available only when developing for Windows.

Gets the offset from the start of the index buffer to the first vertex index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int BaseVertex { get; }
```

Property Value

The offset from the start of the index buffer to the first vertex index.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent.Material Property

Note

This property is available only when developing for Windows.

Gets the material of this mesh part.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public MaterialContent Material { get; set; }
```

Property Value

The material of this mesh part.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent.NumVertices Property

Note

This property is available only when developing for Windows.

Gets the number of vertices used in this mesh part.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int NumVertices { get; }
```

Property Value

The number of vertices used in this mesh part.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent.PrimitiveCount Property

Note

This property is available only when developing for Windows.

Gets the number of primitives to render for this mesh part.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int PrimitiveCount { get; }
```

Property Value

The number of primitives in this mesh part.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent.StartIndex Property

Note

This property is available only when developing for Windows.

Gets the location in the index buffer at which to start reading vertices.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int StartIndex { get; }
```

Property Value

The location in the index buffer at which to start reading vertices.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent.StreamOffset Property

Note

This property is available only when developing for Windows.

Gets the offset, in bytes, from the first index of the of vertex buffer for this mesh part.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public int StreamOffset { get; }
```

Property Value

The offset, in bytes, from the first index of the of vertex buffer for this mesh part.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContent.Tag Property

Note

This property is available only when developing for Windows.

Gets a user-defined tag object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Object Tag { get; set; }
```

Property Value

A user-defined tag object.

See Also

Reference

[ModelMeshPartContent Class](#)

[ModelMeshPartContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelMeshPartContentCollection Class

Note

This class is available only when developing for Windows.

Collection of [ModelMeshPartContent](#) objects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ModelMeshPartContentCollection : ReadOnlyCollection<ModelMeshPartContent>
```

See Also

Reference

[ModelMeshPartContentCollection Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

ModelMeshPartContentCollection Members

The following tables list the members exposed by the ModelMeshPartContentCollection type.










Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)



Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also










Reference

[ModelMeshPartContentCollection Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshPartContentCollection Methods

Public Methods

	Name	Description
	Contains	(Inherited from ReadOnlyCollection.)
	CopyTo	(Inherited from ReadOnlyCollection.)
	Equals	(Inherited from Object.)
	GetEnumerator	(Inherited from ReadOnlyCollection.)
	GetHashCode	(Inherited from Object.)
	GetType	(Inherited from Object.)
	IndexOf	(Inherited from ReadOnlyCollection.)
	ReferenceEquals	(Inherited from Object.)
	ToString	(Inherited from Object.)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object.)
	MemberwiseClone	(Inherited from Object.)

See Also



Reference

[ModelMeshPartContentCollection Class](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelMeshPartContentCollection Properties

Public Properties

	Name	Description
	Count	(Inherited from ReadOnlyCollection.)
	Item	(Inherited from ReadOnlyCollection.)

Protected Properties

	Name	Description
	Items	(Inherited from ReadOnlyCollection.)

See Also

Reference

[ModelMeshPartContentCollection Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelProcessor Class

Note

This class is available only when developing for Windows.

Processes a game asset mesh to a model content that is optimal for runtime.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class ModelProcessor : ContentProcessor<NodeContent, ModelContent>
```

See Also

Reference

[ModelProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista












ModelProcessor Members

The following tables list the members exposed by the ModelProcessor type.







Public Constructors

Name	Description
 ModelProcessor	Initializes a new instance of ModelProcessor.






Public Properties

Name	Description
 ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
 ColorKeyEnabled	Specifies whether color keying of a model is enabled.
 GenerateMipmaps	Specifies whether a full chain of mipmaps is generated from the source material.
 GenerateTangentFrames	Gets or sets the value of the Generate Tangent Frames processor parameter.
 ResizeTexturesToPowerOfTwo	Specifies whether resizing of textures are enabled.
 RotationX	Gets or sets the value of the X Axis Rotation processor parameter.
 RotationY	Gets or sets the value of the Y Axis Rotation processor parameter.
 RotationZ	Gets or sets the value of the Z Axis Rotation processor parameter.
 Scale	Gets or sets the value of the Scale processor parameter.
 SwapWindingOrder	Gets or sets the value of the Swap Winding Order processor parameter.
 TextureFormat	Specifies the texture format of output materials.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Converts mesh content to model content.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 ConvertMaterial	Called by the framework when the MaterialContent property of a GeometryContent object is encountered in the input node collection.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 ProcessGeometryUsingMaterial	Processes all geometry using a specified material.
 ProcessVertexChannel	Processes geometry content vertex channels at the specified index.

See Also

Reference

[ModelProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **ModelProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelProcessor ()
```

See Also

Reference

[ModelProcessor Class](#)







[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)






Platforms Windows XP SP2, Windows Vista

ModelProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Converts mesh content to model content.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	ConvertMaterial	Called by the framework when the MaterialContent property of a GeometryContent object is encountered in the input node collection.
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	ProcessGeometryUsingMaterial	Processes all geometry using a specified material.
	ProcessVertexChannel	Processes geometry content vertex channels at the specified index.

See Also

Reference

[ModelProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelProcessor.ConvertMaterial Method

Note

This method is available only when developing for Windows.

Called by the framework when the [MaterialContent](#) property of a [GeometryContent](#) object is encountered in the input node collection.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual MaterialContent ConvertMaterial (  
    MaterialContent material,  
    ContentProcessorContext context  
)
```

Parameters

material

The input material content.

context

Context for the specified processor.

Return Value

The converted material content.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.Process Method

Note

This method is available only when developing for Windows.

Converts mesh content to model content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override ModelContent Process (
    NodeContent input,
    ContentProcessorContext context
)
```

Parameters

input

The root node content.

context

Context for the specified processor.

Return Value

The model content.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

ModelProcessor.ProcessGeometryUsingMaterial Method

Note

This method is available only when developing for Windows.

Processes all geometry using a specified material.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual void ProcessGeometryUsingMaterial (
    MaterialContent material,
    IEnumerable<GeometryContent> geometryCollection,
    ContentProcessorContext context
)
```

Parameters

material

A material used in the scene.

geometryCollection

A collection of all the geometry using the specified material.

context

Context for the specified processor.

Remarks

This function will be called for each material used by geometry in the input mesh, passing in a collection of all the geometry that is using the specified material. The default implementation replaces **null** materials with a default [BasicMaterialContent](#), validates that the geometry contains texture coordinates if a [BasicMaterialContent](#) has a texture, and sets the [VertexColorEnabled](#) property depending on whether the geometry contains vertex color data.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.ProcessVertexChannel Method

Note

This method is available only when developing for Windows.

Processes geometry content vertex channels at the specified index.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual void ProcessVertexChannel (
    GeometryContent geometry,
    int vertexChannelIndex,
    ContentProcessorContext context
)
```

Parameters

geometry

The geometry content to process.

vertexChannelIndex

Index of the vertex channel to process.

context

Context for the specified processor.

Remarks This function will be called for each [VertexChannel](#) of [Vertices](#) found in the input mesh. Subclasses of [ModelProcessor](#) can override **ProcessVertexChannel** to control how vertex data is processed. The default implementation converts [VertexElementUsage.Color](#) channels to [Color](#) format, and replaces [Weights](#) channels with a pair of [VertexElementUsage.BlendIndices](#) and [VertexElementUsage.BlendIndices](#) channels (using [Byte4](#) format for the [VertexElementUsage.BlendWeight](#), Color for the [VertexElementUsage.BlendWeight](#), and discarding excess weights if there are more than four per vertex).

See Also

Reference

[ModelProcessor Class](#)












[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

ModelProcessor Properties

Public Properties

	Name	Description
	ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
	ColorKeyEnabled	Specifies whether color keying of a model is enabled.
	GenerateMipmaps	Specifies whether a full chain of mipmaps is generated from the source material.
	GenerateTangentFrames	Gets or sets the value of the Generate Tangent Frames processor parameter.
	ResizeTexturesToPowerOfTwo	Specifies whether resizing of textures are enabled.
	RotationX	Gets or sets the value of the X Axis Rotation processor parameter.
	RotationY	Gets or sets the value of the Y Axis Rotation processor parameter.
	RotationZ	Gets or sets the value of the Z Axis Rotation processor parameter.
	Scale	Gets or sets the value of the Scale processor parameter.
	SwapWindingOrder	Gets or sets the value of the Swap Winding Order processor parameter.
	TextureFormat	Specifies the texture format of output materials.

Remarks

For more information about creating or setting processor parameters, see [Parameterized Processors](#).

See Also

Reference

[ModelProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelProcessor.ColorKeyColor Property

Note

This property is available only when developing for Windows.

Specifies the color used when color keying for a texture is enabled. When color keying, all pixels of a specified color are replaced with transparent black.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual Color ColorKeyColor { get; set; }
```

Property Value

Color value of the material to replace with transparent black.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.ColorKeyEnabled Property

Note

This property is available only when developing for Windows.

Specifies whether color keying of a model is enabled.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ColorKeyEnabled { get; set; }
```

Property Value

true if color keying is enabled; **false** otherwise.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.GenerateMipmaps Property

Note

This property is available only when developing for Windows.

Specifies whether a full chain of mipmaps is generated from the source material. Existing mipmaps of the material are not replaced.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool GenerateMipmaps { get; set; }
```

Property Value

true if mipmap generation is enabled; **false** otherwise.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.GenerateTangentFrames Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the Generate Tangent Frames processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool GenerateTangentFrames { get; set; }
```

Property Value

true if binormals and tangents should be generated if none are found; otherwise **false**.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.ResizeTexturesToPowerOfTwo Property

Note

This property is available only when developing for Windows.

Specifies whether resizing of textures are enabled. Typically used to maximize compatability with a graphics card because many graphics cards do not support a material size that is not a power of two. If **ResizeTexturesToPowerOfTwo** is enabled, textures are resized to the next largest power of two.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ResizeTexturesToPowerOfTwo { get; set; }
```

Property Value

true if resizing is enabled; **false** otherwise.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.RotationX Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the X Axis Rotation processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual float RotationX { get; set; }
```

Property Value

The amount of rotation, in degrees, around the X axis.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.RotationY Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the Y Axis Rotation processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual float RotationY { get; set; }
```

Property Value

The amount of rotation, in degrees, around the X axis.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.RotationZ Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the Z Axis Rotation processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual float RotationZ { get; set; }
```

Property Value

The amount of rotation, in degrees, around the Z axis.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.Scale Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the Scale processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual float Scale { get; set; }
```

Property Value

The scaling factor to be applied.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.SwapWindingOrder Property

Note

This property is available only when developing for Windows.

Gets or sets the value of the Swap Winding Order processor parameter.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool SwapWindingOrder { get; set; }
```

Property Value

true if the winding order of the model should be swapped; otherwise **false**. This is useful for models that appear to be drawn inside-out.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelProcessor.TextureFormat Property

Note

This property is available only when developing for Windows.

Specifies the texture format of output materials. Materials can either be left unchanged from the source asset, converted to a corresponding [Color](#), or compressed using the appropriate **DXTCompressed** format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual TextureProcessorOutputFormat TextureFormat { get; set; }
```

Property Value

The texture format of the output.

See Also

Reference

[ModelProcessor Class](#)

[ModelProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelTextureProcessor Class

Note

This class is available only when developing for Windows.

Processes textures for 3D models.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]  
public class ModelTextureProcessor : TextureProcessor
```

See Also

Reference

[ModelTextureProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista






ModelTextureProcessor Members

The following tables list the members exposed by the ModelTextureProcessor type.






Public Constructors

Name	Description
 ModelTextureProcessor	Initializes a new instance of the ModelTextureProcessor class.



Public Properties

Name	Description
 ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
 ColorKeyEnabled	Specifies whether color keying of the texture is enabled.
 GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source material.
 ResizeToPowerOfTwo	Specifies whether resizing of textures are enabled.
 TextureFormat	Specifies the texture format of output materials.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ModelTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelTextureProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ModelTextureProcessor](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ModelTextureProcessor ()
```

See Also

Reference

[ModelTextureProcessor Class](#)






[ModelTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

ModelTextureProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[ModelTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelTextureProcessor Properties

Public Properties

	Name	Description
	ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
	ColorKeyEnabled	Specifies whether color keying of the texture is enabled.
	GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source material.
	ResizeToPowerOfTwo	Specifies whether resizing of textures are enabled.
	TextureFormat	Specifies the texture format of output materials.

See Also

Reference

[ModelTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

ModelTextureProcessor.ColorKeyColor Property

Note

This property is available only when developing for Windows.

Specifies the color used when color keying for a texture is enabled. When color keying, all pixels of a specified color are replaced with transparent black.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override Color ColorKeyColor { get; }
```

Property Value

Color value of the material to replace with transparent black.

See Also

Reference

[ModelTextureProcessor Class](#)

[ModelTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelTextureProcessor.ColorKeyEnabled Property

Note

This property is available only when developing for Windows.

Specifies whether color keying of the texture is enabled.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool ColorKeyEnabled { get; }
```

Property Value

true if color keying is enabled; **false** otherwise.

See Also

Reference

[ModelTextureProcessor Class](#)

[ModelTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelTextureProcessor.GenerateMipmaps Property

Note

This property is available only when developing for Windows.

Specifies if a full chain of mipmaps are generated from the source material. Existing mipmaps of the material are not replaced.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool GenerateMipmaps { get; }
```

Property Value

true if mipmap generation is enabled; **false** otherwise.

See Also

Reference

[ModelTextureProcessor Class](#)

[ModelTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelTextureProcessor.ResizeToPowerOfTwo Property

Note

This property is available only when developing for Windows.

Specifies whether resizing of textures are enabled. Typically used to maximize compatability with a graphics card because many graphics cards do not support a material size that is not a power of two. If **ResizeToPowerOfTwo** is enabled, textures are resized to the next largest power of two.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool ResizeToPowerOfTwo { get; }
```

Property Value

true if resizing is enabled; **false** otherwise.

See Also

Reference

[ModelTextureProcessor Class](#)

[ModelTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ModelTextureProcessor.TextureFormat Property

Note

This property is available only when developing for Windows.

Specifies the texture format of output materials. Materials can either be left unchanged from the source asset, converted to a corresponding [Color](#), or compressed using the appropriate **DXTCompressed** format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override TextureProcessorOutputFormat TextureFormat { get; }
```

Property Value

The texture format of the output.

See Also

Reference

[ModelTextureProcessor Class](#)

[ModelTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

PassThroughProcessor Class

Note

This class is available only when developing for Windows.

Provides an implementation of a no-operation processor that returns an unmodified copy of the input data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]
public class PassThroughProcessor : ContentProcessor<Object, Object>
```

Remarks **PassThroughProcessor** is useful when the importer is already producing data in a game ready format. This can occur after importing textures from DDS files that have been formatted using an external tool, or when deserializing game logic data from XML using the intermediate XML importer.

See Also

Reference

[PassThroughProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista







PassThroughProcessor Members

The following tables list the members exposed by the PassThroughProcessor type.



Public Constructors

Name	Description
 PassThroughProcessor	Initializes a new instance of the PassThroughProcessor class.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	No-operation process that simply returns the input argument.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PassThroughProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PassThroughProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [PassThroughProcessor](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public PassThroughProcessor ()
```

See Also

Reference

[PassThroughProcessor Class](#)







[PassThroughProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

PassThroughProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	No-operation process that simply returns the input argument.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[PassThroughProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PassThroughProcessor.Process Method

Note

This method is available only when developing for Windows.

No-operation process that simply returns the input argument.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override Object Process (  
    Object input,  
    ContentProcessorContext context  
)
```

Parameters

input

The input argument.

context

Context for the specified processor.

Return Value

The unmodified input argument.

See Also

Reference

[PassThroughProcessor Class](#)

[PassThroughProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

SongContent Class

Note

This class is available only when developing for Windows.

Represents a processed [Song](#) object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class SongContent
```

See Also

Reference

[SongContent Members](#)






[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista



SongContent Members

The following tables list the members exposed by the SongContent type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[SongContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SongContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SongContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SongProcessor Class

Note

This class is available only when developing for Windows.

A custom song processor that processes an intermediate [AudioContent](#) type. This type encapsulates the source audio content, producing a [Song](#) type that can be used in the game.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class SongProcessor : ContentProcessor<AudioContent, SongContent>
```

See Also

Reference

[SongProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista


SongProcessor Members

The following tables list the members exposed by the SongProcessor type.







Public Constructors

	Name	Description
	SongProcessor	Initializes a new instance of SongProcessor.



Public Properties

	Name	Description
	Quality	Gets or sets the target format quality of the audio content.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Builds the content for the source audio.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SongProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SongProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **SongProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public SongProcessor ()
```

See Also

Reference

[SongProcessor Class](#)







[SongProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

SongProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Builds the content for the source audio.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SongProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SongProcessor.Process Method

Note

This method is available only when developing for Windows.

Builds the content for the source audio.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override SongContent Process (  
    AudioContent input,  
    ContentProcessorContext context  
)
```

Parameters

input

The audio content to build.

context

Context for the specified processor.

Return Value

The built audio.

See Also

Reference

[SongProcessor Class](#)


[SongProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

SongProcessor Properties

Public Properties

	Name	Description
	Quality	Gets or sets the target format quality of the audio content.

See Also

Reference

[SongProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SongProcessor.Quality Property

Note

This property is available only when developing for Windows.

Gets or sets the target format quality of the audio content.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ConversionQuality Quality { get; set; }
```

Property Value

The [ConversionQuality](#) of this audio data.

See Also

Reference

[SongProcessor Class](#)

[SongProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SoundEffectContent Class

Note

This class is available only when developing for Windows.

Represents a processed sound effect.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class SoundEffectContent
```

See Also

Reference

[SoundEffectContent Members](#)






[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista



SoundEffectContent Members

The following tables list the members exposed by the SoundEffectContent type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[SoundEffectContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SoundEffectContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SoundEffectContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SoundEffectProcessor Class

Note

This class is available only when developing for Windows.

Provides a custom sound effect processor that processes an intermediate [AudioContent](#) type. This type encapsulates the source audio content, producing a [SoundEffect](#) type that can be used in the game.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class SoundEffectProcessor : ContentProcessor<AudioContent, SoundEffectContent>
```

See Also

Reference

[SoundEffectProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista


SoundEffectProcessor Members

The following tables list the members exposed by the SoundEffectProcessor type.







Public Constructors

	Name	Description
	SoundEffectProcessor	Initializes a new instance of SoundEffectProcessor.



Public Properties

	Name	Description
	Quality	Gets or sets the target format quality of the sound effect.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Builds the content for the source audio.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SoundEffectProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SoundEffectProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **SoundEffectProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public SoundEffectProcessor ()
```

See Also

Reference

[SoundEffectProcessor Class](#)







[SoundEffectProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

SoundEffectProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Builds the content for the source audio.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SoundEffectProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SoundEffectProcessor.Process Method

Note

This method is available only when developing for Windows.

Builds the content for the source audio.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override SoundEffectContent Process (  
    AudioContent input,  
    ContentProcessorContext context  
)
```

Parameters

input

The audio content to build.

context

Context for the specified processor.

Return Value

The built sound effect.

See Also

Reference

[SoundEffectProcessor Class](#)


[SoundEffectProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

SoundEffectProcessor Properties

Public Properties

	Name	Description
	Quality	Gets or sets the target format quality of the sound effect.

See Also

Reference

[SoundEffectProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SoundEffectProcessor.Quality Property

Note

This property is available only when developing for Windows.

Gets or sets the target format quality of the sound effect.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ConversionQuality Quality { get; set; }
```

Property Value

The [ConversionQuality](#) of the sound effect data.

See Also

Reference

[SoundEffectProcessor Class](#)

[SoundEffectProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SpriteFontContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing a design-time [SpriteFont Class](#) holding packed font data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class SpriteFontContent
```

Remarks This class is the output of the built-in font processors, [FontDescriptionProcessor](#) and [FontTextureProcessor](#).

See Also

Reference

[FontDescriptionProcessor Class](#)

[FontTextureProcessor Class](#)

[SpriteFontContent Members](#)






[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista



SpriteFontContent Members

The following tables list the members exposed by the SpriteFontContent type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[SpriteFontContent Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SpriteFontContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SpriteFontContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SpriteTextureProcessor Class

Note

This class is available only when developing for Windows.

Processes textures for use as 2D sprites or user interface components.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]
public class SpriteTextureProcessor : TextureProcessor
```

See Also

Reference

[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista






SpriteTextureProcessor Members

The following tables list the members exposed by the SpriteTextureProcessor type.






Public Constructors

Name	Description
 SpriteTextureProcessor	Initializes a new instance of the SpriteTextureProcessor class.



Public Properties

Name	Description
 ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
 ColorKeyEnabled	Specifies whether color keying of the texture is enabled.
 GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source material.
 ResizeToPowerOfTwo	Specifies whether resizing of textures are enabled.
 TextureFormat	Specifies the texture format of output materials.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[SpriteTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SpriteTextureProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [SpriteTextureProcessor](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public SpriteTextureProcessor ()
```

See Also

Reference

[SpriteTextureProcessor Class](#)






[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

SpriteTextureProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also






Reference

[SpriteTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SpriteTextureProcessor Properties

Public Properties

	Name	Description
	ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
	ColorKeyEnabled	Specifies whether color keying of the texture is enabled.
	GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source material.
	ResizeToPowerOfTwo	Specifies whether resizing of textures are enabled.
	TextureFormat	Specifies the texture format of output materials.

See Also

Reference

[SpriteTextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

SpriteTextureProcessor.ColorKeyColor Property

Note

This property is available only when developing for Windows.

Specifies the color used when color keying for a texture is enabled. When color keying, all pixels of a specified color are replaced with transparent black.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override Color ColorKeyColor { get; }
```

Property Value

Color value of the material to replace with transparent black.

See Also

Reference

[SpriteTextureProcessor Class](#)

[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SpriteTextureProcessor.ColorKeyEnabled Property

Note

This property is available only when developing for Windows.

Specifies whether color keying of the texture is enabled.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool ColorKeyEnabled { get; }
```

Property Value

true if color keying is enabled; **false** otherwise.

See Also

Reference

[SpriteTextureProcessor Class](#)

[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SpriteTextureProcessor.GenerateMipmaps Property

Note

This property is available only when developing for Windows.

Specifies if a full chain of mipmaps are generated from the source material. Existing mipmaps of the material are not replaced.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool GenerateMipmaps { get; }
```

Property Value

true if mipmap generation is enabled; **false** otherwise.

See Also

Reference

[SpriteTextureProcessor Class](#)

[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SpriteTextureProcessor.ResizeToPowerOfTwo Property

Note

This property is available only when developing for Windows.

Specifies whether resizing of textures are enabled. Typically used to maximize compatability with a graphics card because many graphics cards do not support a material size that is not a power of two. If **ResizeToPowerOfTwo** is enabled, textures are resized to the next largest power of two.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[BrowsableAttribute(false)]  
public override bool ResizeToPowerOfTwo { get; }
```

Property Value

true if resizing is enabled; **false** otherwise.

See Also

Reference

[SpriteTextureProcessor Class](#)

[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

SpriteTextureProcessor.TextureFormat Property

Note

This property is available only when developing for Windows.

Specifies the texture format of output materials. Materials can either be left unchanged from the source asset, converted to a corresponding [Color](#), or compressed using the appropriate **DXTCompressed** format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override TextureProcessorOutputFormat TextureFormat { get; }
```

Property Value

The texture format of the output.

See Also

Reference

[SpriteTextureProcessor Class](#)

[SpriteTextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureProcessor Class

Note

This class is available only when developing for Windows.

Provides methods for processing textures.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[ContentProcessorAttribute]  
public class TextureProcessor : ContentProcessor<TextureContent, TextureContent>
```

See Also

Reference

[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista






TextureProcessor Members

The following tables list the members exposed by the TextureProcessor type.







Public Constructors

Name	Description
 TextureProcessor	Initializes a new instance of the TextureProcessor class.



Public Properties

Name	Description
 ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
 ColorKeyEnabled	Specifies whether color keying of a texture is enabled.
 GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source texture.
 ResizeToPowerOfTwo	Specifies whether resizing of a texture is enabled.
 TextureFormat	Specifies the texture format of outputmaterials.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Processes a texture.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

TextureProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [TextureProcessor](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TextureProcessor ()
```

See Also

Reference

[TextureProcessor Class](#)







[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

TextureProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Processes a texture.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[TextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

TextureProcessor.Process Method

Note

This method is available only when developing for Windows.

Processes a texture.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override TextureContent Process (  
    TextureContent input,  
    ContentProcessorContext context  
)
```

Parameters

input

The texture content to process.

context

Context for the specified processor.

Return Value

The converted texture content.

Remarks **Process** generates mipmaps if they are not already present, but leaves the texture in the same format as the input data. This is suitable for use with DDS files that have already been converted to an appropriate format.

See Also

Reference

[TextureProcessor Class](#)






[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureProcessor Properties

Public Properties

	Name	Description
	ColorKeyColor	Specifies the color used when color keying for a texture is enabled.
	ColorKeyEnabled	Specifies whether color keying of a texture is enabled.
	GenerateMipmaps	Specifies if a full chain of mipmaps are generated from the source texture.
	ResizeToPowerOfTwo	Specifies whether resizing of a texture is enabled.
	TextureFormat	Specifies the texture format of outputmaterials.

See Also

Reference

[TextureProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

TextureProcessor.ColorKeyColor Property

Note

This property is available only when developing for Windows.

Specifies the color used when color keying for a texture is enabled. When color keying, all pixels of a specified color are replaced with transparent black.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual Color ColorKeyColor { get; set; }
```

Property Value

The color used when a texture is color keyed.

See Also

Reference

[TextureProcessor Class](#)

[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureProcessor.ColorKeyEnabled Property

Note

This property is available only when developing for Windows.

Specifies whether color keying of a texture is enabled.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ColorKeyEnabled { get; set; }
```

Property Value

true if color keying is enabled; **false** otherwise.

See Also

Reference

[TextureProcessor Class](#)

[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureProcessor.GenerateMipmaps Property

Note

This property is available only when developing for Windows.

Specifies if a full chain of mipmaps are generated from the source texture. Existing mipmaps of the texture are not replaced.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool GenerateMipmaps { get; set; }
```

Property Value

true if mipmap generation is enabled; **false** otherwise.

See Also

Reference

[TextureProcessor Class](#)

[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

TextureProcessor.ResizeToPowerOfTwo Property

Note

This property is available only when developing for Windows.

Specifies whether resizing of a texture is enabled. Typically used to maximize compatibility with a graphics card because many graphics cards do not support a texture size that is not a power of two. If **ResizeToPowerOfTwo** is enabled, the texture is resized to the next largest power of two.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ResizeToPowerOfTwo { get; set; }
```

Property Value

true if resizing is enabled; **false** otherwise.

See Also

Reference

[TextureProcessor Class](#)

[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureProcessor.TextureFormat Property

Note

This property is available only when developing for Windows.

Specifies the texture format of outputmaterials. Materials can either be left unchanged from the source asset, converted to a corresponding [Color](#), or compressed using the appropriate **DXTCompressed** format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual TextureProcessorOutputFormat TextureFormat { get; set; }
```

Property Value

The texture format of the output.

See Also

Reference

[TextureProcessor Class](#)

[TextureProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

TextureProcessorOutputFormat Enumeration

Note

This enumeration is available only when developing for Windows.

Specifies the target output (of type [SurfaceFormat](#)) of the texture processor. Used by [TextureProcessor.TextureFormat](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public enum TextureProcessorOutputFormat
```

Members

Member name	Description
Color	The SurfaceFormat value, of the input TextureContent object, is converted to Color by the processor. Typically used for 2D graphics and overlays.
DxtCompressed	The SurfaceFormat value, of the input TextureContent object, is converted to an appropriate DXT compression by the processor. If the input texture contains fractional alpha values, it is converted to DXT5 format (8 bits per texel); otherwise it is converted to DXT1 (4 bits per texel). This conversion reduces the resource's size on the graphics card. Typically used for 3D textures such as 3D model textures.
NoChange	The SurfaceFormat value, of the input TextureContent object, is not changed by the processor. Typically used for textures processed by an external tool.

See Also

Reference

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for managing a design-time vertex buffer that holds packed vertex data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class VertexBufferContent : ContentItem
```

Remarks

This type directly corresponds to the runtime [VertexBuffer](#) class, and when a **VertexBufferContent** object is passed to the content compiler, the vertex data deserializes directly into a [VertexBuffer](#) at runtime. **VertexBufferContent** objects are not directly created by importers. The preferred method is to store vertex data in the more flexible [VertexContent](#) class.

See Also

Reference

[VertexBufferContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista





VertexBufferContent Members

The following tables list the members exposed by the VertexBufferContent type.









Public Constructors

	Name	Description
	VertexBufferContent	Overloaded. Initializes a new instance of VertexBufferContent.



Public Properties

	Name	Description
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	VertexData	Gets the array containing the raw bytes of the packed vertex data.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	 SizeOf	Gets the size of the specified type, in bytes.
	ToString	(Inherited from Object .)
	Write	Overloaded. Writes additional data into the vertex buffer.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexBufferContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VertexBufferContent Constructor

Initializes a new instance of **VertexBufferContent**.

Overload List

Name	Description
VertexBufferContent ()	Initializes a new instance of VertexBufferContent .
VertexBufferContent (Int32)	Initializes a new instance of VertexBufferContent of the specified size.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VertexBufferContent Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **VertexBufferContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexBufferContent ()
```

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent Constructor (Int32)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **VertexBufferContent** of the specified size.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VertexBufferContent (  
    int size  
)
```

Parameters

size

The size of the vertex buffer content, in bytes.

See Also

Reference

[VertexBufferContent Class](#)









[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

VertexBufferContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	 SizeOf	Gets the size of the specified type, in bytes.
	ToString	(Inherited from Object .)
	Write	Overloaded. Writes additional data into the vertex buffer.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VertexBufferContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VertexBufferContent.SizeOf Method

Note

This method is available only when developing for Windows.

Gets the size of the specified type, in bytes.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static int SizeOf (  
    Type type  
)
```

Parameters

type

The type.

Return Value

The size of the specified type, in bytes.

Remarks

Call this method to compute offset parameters for the [Write](#) method. If the specified data type cannot be packed into a vertex buffer—for example, if *type* is not a valid value type—a [NotSupportedException](#) is thrown.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent.Write Method

Writes additional data into the vertex buffer.

Overload List

Name	Description
VertexBufferContent.Write (Int32, Int32, Generic IEnumerable)	Writes additional data into the vertex buffer.
VertexBufferContent.Write (Int32, Int32, Generic IEnumerable, TargetPlatform)	Writes additional data into the vertex buffer.
VertexBufferContent.Write (Int32, Int32, Type, IEnumerable)	Writes additional data into the vertex buffer.
VertexBufferContent.Write (Int32, Int32, Type, IEnumerable, TargetPlatform)	Writes additional data into the vertex buffer.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VertexBufferContent.Write Generic Method (Int32, Int32, Generic IEnumerable)

Note

This generic method is available only when developing for Windows.

Writes additional data into the vertex buffer. Writing begins at the specified byte offset, and each value is spaced according to the specified stride value (in bytes).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write<T> (  
    int offset,  
    int stride,  
    IEnumerable<T> data  
)
```

Type Parameters

T

Type being written.

Parameters

offset

Offset to begin writing at.

stride

Stride of the data being written, in bytes.

data

Enumerated collection of data.

Remarks

This method automatically grows the vertex buffer if an attempt is made to write past the buffer end. **Write** throws [NotSupportedException](#) if the specified data type cannot be packed into a vertex buffer. For example, if *data* is not a valid value type.

Use this method to interleave vertex data channels into a single buffer. This can be done by passing the total vertex size as the *stride* and suitable smaller offsets for each channel. You can also concatenate entire vertex buffers by passing the length of the vertex as the *offset*, 1 as the *stride*, and the vertex data as the *data* parameter.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent.Write Generic Method (Int32, Int32, Generic IEnumerable, TargetPlatform)

Note

This generic method is available only when developing for Windows.

Writes additional data into the vertex buffer. Writing begins at the specified byte offset, and each value is spaced according to the specified stride value (in bytes).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write<T> (  
    int offset,  
    int stride,  
    IEnumerable<T> data,  
    TargetPlatform targetPlatform  
)
```

Type Parameters

T

Type of value being written.

Parameters

offset

Offset at which to begin writing.

stride

Stride of the data being written, in bytes.

data

The data to write.

targetPlatform

The target platform of the content build.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent.Write Method (Int32, Int32, Type, IEnumerable)

Note

This method is available only when developing for Windows.

Writes additional data into the vertex buffer. Writing begins at the specified byte offset, and each value is spaced according to the specified stride value (in bytes).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    int offset,  
    int stride,  
    Type dataType,  
    IEnumerable data  
)
```

Parameters

offset

Offset at which to begin writing.

stride

Stride of the data being written, in bytes.

dataType

The type of data to be written.

data

The data to write.

Remarks

This method automatically grows the vertex buffer if an attempt is made to write past the buffer end. **Write** throws [NotSupportedException](#) if the specified data type cannot be packed into a vertex buffer. For example, if *data* is not a valid value type.

Use this method to interleave vertex data channels into a single buffer. This can be done by passing the total vertex size as the *stride* and suitable smaller offsets for each channel. You can also concatenate entire vertex buffers by passing the length of the vertex as the *offset*, 1 as the *stride*, and the vertex data as the *data* parameter.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent.Write Method (Int32, Int32, Type, IEnumerable, TargetPlatform)

Note

This method is available only when developing for Windows.

Writes additional data into the vertex buffer. Writing begins at the specified byte offset, and each value is spaced according to the specified stride value (in bytes).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    int offset,  
    int stride,  
    Type dataType,  
    IEnumerable data,  
    TargetPlatform targetPlatform  
)
```

Parameters

offset

Offset at which to begin writing.

stride

Stride of the data being written, in bytes.

dataType

The type of data to be written.

data

The data to write.

targetPlatform

The target platform of the content build.

See Also

Reference

[VertexBufferContent Class](#)





[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VertexBufferContent Properties

Public Properties

	Name	Description
	Identity	(Inherited from ContentItem .)
	Name	(Inherited from ContentItem .)
	OpaqueData	(Inherited from ContentItem .)
	VertexData	Gets the array containing the raw bytes of the packed vertex data.

See Also

Reference

[VertexBufferContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VertexBufferContent.VertexData Property

Note

This property is available only when developing for Windows.

Gets the array containing the raw bytes of the packed vertex data. Use this method to get and set the contents of the vertex buffer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public byte[] VertexData { get; }
```

Property Value

Raw data of the packed vertex data.

See Also

Reference

[VertexBufferContent Class](#)

[VertexBufferContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista

VideoProcessor Class

Note

This class is available only when developing for Windows.

Processes videos for playback within a game.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class VideoProcessor : ContentProcessor<VideoContent, VideoContent>
```

See Also

Reference

[VideoProcessor Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista


VideoProcessor Members

The following tables list the members exposed by the VideoProcessor type.







Public Constructors

Name	Description
 VideoProcessor	Initializes a new instance of VideoProcessor.



Public Properties

Name	Description
 VideoSoundtrackType	Gets or sets the VideoSoundtrackType for this video.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 Process	Processes a video.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VideoProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VideoProcessor Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **VideoProcessor**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VideoProcessor ()
```

See Also

Reference

[VideoProcessor Class](#)







[VideoProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)



Platforms Windows XP SP2, Windows Vista

VideoProcessor Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	Process	Processes a video.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[VideoProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VideoProcessor.Process Method

Note

This method is available only when developing for Windows.

Processes a video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override VideoContent Process (  
    VideoContent input,  
    ContentProcessorContext context  
)
```

Parameters

input

The video content to process.

context

Context for the processor.

Return Value

The converted video content.

See Also

Reference

[VideoProcessor Class](#)


[VideoProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

Platforms Windows XP SP2, Windows Vista

VideoProcessor Properties

Public Properties

	Name	Description
	VideoSoundtrackType	Gets or sets the VideoSoundtrackType for this video.

See Also

Reference

[VideoProcessor Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

VideoProcessor.VideoSoundtrackType Property

Note

This property is available only when developing for Windows.

Gets or sets the [VideoSoundtrackType](#) for this video.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Processors

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public VideoSoundtrackType VideoSoundtrackType { get; set; }
```

Property Value

The new [VideoSoundtrackType](#) for the video.

See Also

Reference

[VideoProcessor Class](#)

[VideoProcessor Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Processors Namespace](#)

PlatformsWindows XP SP2, Windows Vista






Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace

Note

This namespace is available only when developing for Windows.

Provides base classes that represent compilers and writers used by the XNA Framework Content Pipeline when processing specific game asset types.

Classes

	Name	Description
	ContentCompiler	Provides methods for writing compiled binary format.
	ContentTypeWriter	Provides a generic implementation of ContentTypeWriter methods and properties for compiling a specific managed type into a binary format.
	ContentTypeWriter	Provides methods and properties for compiling a specific managed type into a binary format.
	ContentTypeWriterAttribute	Identifies the components of a type writer.
	ContentWriter	Provides an implementation for many of the ContentCompiler methods including compilation, state tracking for or shared resources and creation of the header type manifest.

ContentCompiler Class

Note

This class is available only when developing for Windows.

Provides methods for writing compiled binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ContentCompiler
```

See Also

Reference

[ContentCompiler Members](#)







[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista



ContentCompiler Members

The following tables list the members exposed by the ContentCompiler type.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetTypeWriter	Retrieves the worker writer for the specified type.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also







Reference

[ContentCompiler Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentCompiler Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetTypeWriter	Retrieves the worker writer for the specified type.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentCompiler Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentCompiler.GetTypeWriter Method

Note

This method is available only when developing for Windows.

Retrieves the worker writer for the specified type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentTypeWriter GetTypeWriter (  
    Type type  
)
```

Parameters

type

The type.

Return Value

The worker writer.

Remarks This should be called from the [ContentTypeWriter.Initialize](#) method.

See Also

Reference

[Initialize](#)

[ContentCompiler Class](#)

[ContentCompiler Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeWriter Class

Note

This class is available only when developing for Windows.

Provides methods and properties for compiling a specific managed type into a binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentTypeWriter
```

See Also

Reference

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista




ContentTypeWriter Members

The following tables list the members exposed by the ContentTypeWriter type.








Protected Constructors

Name	Description
 ContentTypeWriter	Initializes a new instance of the ContentTypeWriter class.






Public Properties

Name	Description
 CanDeserializeIntoExistingObject	Determines if deserialization into an existing object is possible.
 TargetType	Gets the type handled by this compiler component.
 TypeVersion	Gets a format version number for this type.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetRuntimeReader	Gets the assembly qualified name of the runtime loader for this type.
 GetRuntimeType	Gets the assembly qualified name of the runtime target type.
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 Initialize	Retrieves and caches nested type writers and allows for reflection over the target data type.
 MemberwiseClone	(Inherited from Object .)
 ShouldCompressContent	Indicates whether a given type of content should be compressed.
 Write	Compiles an object into binary format.

See Also

Reference

[ContentTypeWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeWriter](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentTypeWriter (  
    Type targetType  
)
```

Parameters

targetType

The target type the [ContentTypeWriter](#).

Exceptions

Exception type	Condition
ArgumentException	<ul style="list-style-type: none">Cannot serialize type <i>targetType</i>. Pointers and references are not supported.<i>targetType</i> cannot be serialized because not all the generic type parameters have been specified.
ArgumentNullException	<i>targetType</i> is null .

See Also

Reference

[ContentTypeWriter Class](#)








[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)






PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetRuntimeReader	Gets the assembly qualified name of the runtime loader for this type.
	GetRuntimeType	Gets the assembly qualified name of the runtime target type.
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	Initialize	Retrieves and caches nested type writers and allows for reflection over the target data type.
	MemberwiseClone	(Inherited from Object .)
	ShouldCompressContent	Indicates whether a given type of content should be compressed.
	Write	Compiles an object into binary format.

See Also

Reference

[ContentTypeWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriter.GetRuntimeReader Method

Note

This method is available only when developing for Windows.

Gets the assembly qualified name of the runtime loader for this type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract string GetRuntimeReader (  
    TargetPlatform targetPlatform  
)
```

Parameters

targetPlatform

Name of the platform.

Return Value

Name of the runtime loader.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.GetRuntimeType Method

Note

This method is available only when developing for Windows.

Gets the assembly qualified name of the runtime target type. The runtime target type often matches the design time type, but may differ.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual string GetRuntimeType (  
    TargetPlatform targetPlatform  
)
```

Parameters

targetPlatform

The target platform.

Return Value

The qualified name.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.Initialize Method

Note

This method is available only when developing for Windows.

Retrieves and caches nested type writers and allows for reflection over the target data type. Called by the framework at creation time.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected virtual void Initialize (  
    ContentCompiler compiler  
)
```

Parameters

compiler

The content compiler.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.ShouldCompressContent Method

Note

This method is available only when developing for Windows.

Indicates whether a given type of content should be compressed.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal virtual bool ShouldCompressContent (
    TargetPlatform targetPlatform,
    Object value
)
```

Parameters

targetPlatform

The target platform of the content build.

value

The object about to be serialized, or **null** if a collection of objects is to be serialized.

Return Value

true if the content of the requested type should be compressed; **false** otherwise.

Remarks

This base class implementation of this method always returns **true**. It should be overridden to return **false** if there would be little or no useful reduction in size of the content type's data from a general-purpose lossless compression algorithm.

The implementations for [Song Class](#) and [SoundEffect Class](#) data return **false** because data for these content types is already in compressed form.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.Write Method

Note

This method is available only when developing for Windows.

Compiles an object into binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal abstract void Write (  
    ContentTypeWriter output,  
    Object value  
)
```

Parameters

output

The content writer serializing the value.

value

The resultant object.

See Also

Reference

[ContentTypeWriter Class](#)




[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter Properties

Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	Determines if deserialization into an existing object is possible.
	TargetType	Gets the type handled by this compiler component.
	TypeVersion	Gets a format version number for this type.

See Also

Reference

[ContentTypeWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriter.CanDeserializeIntoExistingObject Property

Note

This property is available only when developing for Windows.

Determines if deserialization into an existing object is possible.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool CanDeserializeIntoExistingObject { get; }
```

Property Value

true if the object can be deserialized into; **false** otherwise.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.TargetType Property

Note

This property is available only when developing for Windows.

Gets the type handled by this compiler component.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Type TargetType { get; }
```

Property Value

The type handled by this compiler component.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.TypeVersion Property

Note

This property is available only when developing for Windows.

Gets a format version number for this type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual int TypeVersion { get; }
```

Property Value

A format version number for this type.

See Also

Reference

[ContentTypeWriter Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter Generic Class

Note

This generic class is available only when developing for Windows.

Provides a generic implementation of [ContentTypeWriter](#) methods and properties for compiling a specific managed type into a binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentTypeWriter<T> : ContentTypeWriter
```

Remarks This is a generic implementation of [ContentTypeWriter](#) and, therefore, can handle strongly typed content data.

See Also

Reference

[ContentTypeWriter](#)

[ContentTypeWriter Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista




ContentTypeWriter Members

The following tables list the members exposed by the ContentTypeWriter type.








Protected Constructors

Name	Description
 ContentTypeWriter	Initializes a new instance of the ContentTypeWriter class.




Public Properties

Name	Description
 CanDeserializeIntoExistingObject	(Inherited from ContentTypeWriter .)
 TargetType	(Inherited from ContentTypeWriter .)
 TypeVersion	(Inherited from ContentTypeWriter .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetRuntimeType	(Inherited from ContentTypeWriter .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)
 Write	Overloaded. Compiles a strongly typed object into binary format.

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 Initialize	(Inherited from ContentTypeWriter .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeWriter Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriter Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeWriter](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentTypeWriter ()
```

See Also

Reference

[ContentTypeWriter Generic Class](#)








[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)




PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetRuntimeType	(Inherited from ContentTypeWriter .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	Write	Overloaded. Compiles a strongly typed object into binary format.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	Initialize	(Inherited from ContentTypeWriter .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeWriter Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriter.Write Method

Compiles a strongly typed object into binary format.

Overload List

Name	Description
ContentTypeWriter.Write (ContentWriter, Object)	Compiles a strongly typed object into binary format.
ContentTypeWriter.Write (ContentWriter, T)	Compiles a strongly typed object into binary format.

See Also

Reference

[ContentTypeWriter Generic Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriter.Write Method (ContentWriter, Object)

Note

This method is available only when developing for Windows.

Compiles a strongly typed object into binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal override void Write (  
    ContentWriter output,  
    Object value  
)
```

Parameters

output

The content writer serializing the value.

value

The value to write.

See Also

Reference

[ContentTypeWriter Generic Class](#)

[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter.Write Method (ContentWriter, T)

Note

This method is available only when developing for Windows.

Compiles a strongly typed object into binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal abstract void Write (  
    ContentWriter output,  
    T value  
)
```

Parameters

output

The content writer serializing the value.

value

The value to write.

See Also

Reference

[ContentTypeWriter Generic Class](#)




[ContentTypeWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriter Properties

Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	(Inherited from ContentTypeWriter .)
	TargetType	(Inherited from ContentTypeWriter .)
	TypeVersion	(Inherited from ContentTypeWriter .)

See Also

Reference

[ContentTypeWriter](#) Generic Class

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler](#) Namespace

ContentTypeWriterAttribute Class

Note

This class is available only when developing for Windows.

Identifies the components of a type writer. Custom content writers must apply this attribute to their class as well as extend the [ContentTypeWriter](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[AttributeUsageAttribute(4)]  
public sealed class ContentTypeWriterAttribute : Attribute
```

See Also

Reference

[ContentTypeWriterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeWriterAttribute Members

The following tables list the members exposed by the ContentTypeWriterAttribute type.










Public Constructors

Name	Description
 ContentTypeWriterAttribute	Initializes a new instance of the ContentTypeWriterAttribute class.



Public Properties

Name	Description
 TypeId	(Inherited from Attribute .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetCustomAttribute	(Inherited from Attribute .)
 GetCustomAttributes	(Inherited from Attribute .)
 GetType	(Inherited from Object .)
 IsDefaultAttribute	(Inherited from Attribute .)
 IsDefined	(Inherited from Attribute .)
 Match	(Inherited from Attribute .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeWriterAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriterAttribute Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeWriterAttribute](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentTypeWriterAttribute ()
```

See Also

Reference

[ContentTypeWriterAttribute Class](#)










[ContentTypeWriterAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentTypeWriterAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also


Reference

[ContentTypeWriterAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentTypeWriterAttribute Properties

Public Properties

	Name	Description
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentTypeWriterAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter Class

Note

This class is available only when developing for Windows.

Provides an implementation for many of the [ContentCompiler](#) methods including compilation, state tracking for shared resources and creation of the header type manifest.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class ContentWriter : BinaryWriter
```

RemarksA new **ContentWriter** is constructed for each compilation operation.

See Also

Reference

[ContentWriter Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista














ContentWriter Members

The following tables list the members exposed by the ContentWriter type.





Public Properties

Name	Description
 BaseStream	(Inherited from BinaryWriter .)
 TargetPlatform	Gets the content build target platform.

Public Methods

Name	Description
 Close	(Inherited from BinaryWriter .)
 Equals	(Inherited from Object .)
 Flush	(Inherited from BinaryWriter .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 Seek	(Inherited from BinaryWriter .)
 ToString	(Inherited from Object .)
 Write	Overloaded. Writes a value.
 WriteExternalReference	Writes the name of an external file to the output binary.
 WriteObject	Overloaded. Writes a single object preceded by a type identifier to the output binary.
 WriteRawObject	Overloaded. Writes a single object to the output binary as an instance of the specified type.
 WriteSharedResource	Adds a shared reference to the output binary and records the object to be serialized later.

Protected Methods

Name	Description
 Dispose	Releases the resources used by the Dispose class.
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)
 Write7BitEncodedInt	(Inherited from BinaryWriter .)

See Also

Reference

[ContentWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter Fields

See Also














Reference

[ContentWriter Class](#)





[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter Methods

Public Methods

	Name	Description
	Close	(Inherited from BinaryWriter .)
	Equals	(Inherited from Object .)
	Flush	(Inherited from BinaryWriter .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	Seek	(Inherited from BinaryWriter .)
	ToString	(Inherited from Object .)
	Write	Overloaded. Writes a value.
	WriteExternalReference	Writes the name of an external file to the output binary.
	WriteObject	Overloaded. Writes a single object preceded by a type identifier to the output binary.
	WriteRawObject	Overloaded. Writes a single object to the output binary as an instance of the specified type.
	WriteSharedResource	Adds a shared reference to the output binary and records the object to be serialized later.

Protected Methods

	Name	Description
	Dispose	Releases the resources used by the Dispose class.
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)
	Write7BitEncodedInt	(Inherited from BinaryWriter .)

See Also

Reference

[ContentWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter.Dispose Method

Note

This method is available only when developing for Windows.

Releases the resources used by the **Dispose** class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected override void Dispose (  
    bool disposing  
)
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.Write Method

Writes a value.

Overload List

Name	Description
ContentWriter.Write (Color)	Writes a Color value.
ContentWriter.Write (Matrix)	Writes a Matrix value.
ContentWriter.Write (Quaternion)	Writes a Quaternion value.
ContentWriter.Write (Vector2)	Writes a Vector2 value.
ContentWriter.Write (Vector3)	Writes a Vector3 value.
ContentWriter.Write (Vector4)	Writes a Vector4 value.
ContentWriter.Write (Boolean)	(Inherited from BinaryWriter .)
ContentWriter.Write (Byte)	(Inherited from BinaryWriter .)
ContentWriter.Write (Byte)	(Inherited from BinaryWriter .)
ContentWriter.Write (Byte, Int32, Int32)	(Inherited from BinaryWriter .)
ContentWriter.Write (Char)	(Inherited from BinaryWriter .)
ContentWriter.Write (Char)	(Inherited from BinaryWriter .)
ContentWriter.Write (Char, Int32, Int32)	(Inherited from BinaryWriter .)
ContentWriter.Write (Decimal)	(Inherited from BinaryWriter .)
ContentWriter.Write (Double)	(Inherited from BinaryWriter .)
ContentWriter.Write (Int16)	(Inherited from BinaryWriter .)
ContentWriter.Write (Int32)	(Inherited from BinaryWriter .)
ContentWriter.Write (Int64)	(Inherited from BinaryWriter .)
ContentWriter.Write (SByte)	(Inherited from BinaryWriter .)
ContentWriter.Write (Single)	(Inherited from BinaryWriter .)
ContentWriter.Write (String)	(Inherited from BinaryWriter .)
ContentWriter.Write (UInt16)	(Inherited from BinaryWriter .)
ContentWriter.Write (UInt32)	(Inherited from BinaryWriter .)
ContentWriter.Write (UInt64)	(Inherited from BinaryWriter .)

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter.Write Method (Color)

Note

This method is available only when developing for Windows.

Writes a [Color](#) value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    Color value  
)
```

Parameters

value

Value of a color using Red, Green, Blue, and Alpha values to write.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.Write Method (Matrix)

Note

This method is available only when developing for Windows.

Writes a [Matrix](#) value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    Matrix value  
)
```

Parameters

value

Value to write.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.Write Method (Quaternion)

Note

This method is available only when developing for Windows.

Writes a [Quaternion](#) value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    Quaternion value  
)
```

Parameters

value

Value to write.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.Write Method (Vector2)

Note

This method is available only when developing for Windows.

Writes a [Vector2](#) value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    Vector2 value  
)
```

Parameters

value

Value to write.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.Write Method (Vector3)

Note

This method is available only when developing for Windows.

Writes a [Vector3](#) value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    Vector3 value  
)
```

Parameters

value

Value to write.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.Write Method (Vector4)

Note

This method is available only when developing for Windows.

Writes a [Vector4](#) value.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void Write (  
    Vector4 value  
)
```

Parameters

value

Value to write.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.WriteExternalReference Generic Method

Note

This generic method is available only when developing for Windows.

Writes the name of an external file to the output binary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteExternalReference<T> (  
    ExternalReference<T> reference  
)
```

Type Parameters

T

The type of *reference*.

Parameters

reference

External reference to a data file for the content item.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentWriter.WriteObject Method

Writes a single object preceded by a type identifier to the output binary.

Overload List

Name	Description
ContentWriter.WriteObject (T)	Writes a single object preceded by a type identifier to the output binary.
ContentWriter.WriteObject (T, ContentType Writer)	Writes a single object to the output binary, using the specified type hint and writer worker.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter.WriteObject Generic Method (T)

Note

This generic method is available only when developing for Windows.

Writes a single object preceded by a type identifier to the output binary.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteObject<T> (  
    T value  
)
```

Type Parameters

T

The type of *value*.

Parameters

value

The value to write.

Remarks This method can be called recursively with a **null** value.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentWriter.WriteObject Generic Method (T, ContentTypeWriter)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output binary, using the specified type hint and writer worker.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteObject<T> (  
    T value,  
    ContentTypeWriter typeWriter  
)
```

Type Parameters

T

The type of *value*.

Parameters

value

The value to write.

typeWriter

The content type writer.

RemarksThe type hint should be retrieved from the [Initialize](#) method of the [ContentTypeWriter](#) that is calling **WriteObject**, by calling [GetTypeWriter](#) and passing it the type of the field used to hold the value being serialized. If the hint type is a sealed value type (which cannot be **null** or hold a polymorphic object instance) this method skips writing the usual type identifier.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentWriter.WriteRawObject Method

Writes a single object to the output binary as an instance of the specified type.

Overload List

Name	Description
ContentWriter.WriteRawObject (T)	Writes a single object to the output binary as an instance of the specified type.
ContentWriter.WriteRawObject (T, ContentTypeWriter)	Writes a single object to the output binary using the specified writer worker.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter.WriteRawObject Generic Method (T)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output binary as an instance of the specified type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteRawObject<T> (  
    T value  
)
```

Type Parameters

T

The type of *value*.

Parameters

value

The value to write.

RemarksIf you specify a base class of the actual object value only data from this base type will be written. This method does not write any type identifier so it cannot support **null** or polymorphic values, and the reader must specify an identical type while loading the compiled data.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentWriter.WriteRawObject Generic Method (T, ContentTypeWriter)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output binary using the specified writer worker.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteRawObject<T> (  
    T value,  
    ContentTypeWriter typeWriter  
)
```

Type Parameters

T

The type of *value*.

Parameters

value

The value to write.

typeWriter

The writer worker. This should be looked up from the [Initialize](#) method of the [ContentTypeWriter](#) that is calling **WriteRawObject**, by calling [GetTypeWriter](#).

Remarks **WriteRawObject** does not write any type identifier, so it cannot support **null** or polymorphic values, and the reader must specify an identical type while loading the compiled data.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentWriter.WriteSharedResource Generic Method

Note

This generic method is available only when developing for Windows.

Adds a shared reference to the output binary and records the object to be serialized later.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteSharedResource<T> (  
    T value  
)
```

Type Parameters

T

The type of *value*.

Parameters

value

The object to record.

See Also

Reference

[ContentWriter Class](#)



[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentWriter Properties

Public Properties

	Name	Description
	BaseStream	(Inherited from BinaryWriter .)
	TargetPlatform	Gets the content build target platform.

See Also

Reference

[ContentWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

ContentWriter.TargetPlatform Property

Note

This property is available only when developing for Windows.

Gets the content build target platform.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public TargetPlatform TargetPlatform { get; }
```

Property Value

The content build target platform.

See Also

Reference

[ContentWriter Class](#)

[ContentWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Compiler Namespace](#)

PlatformsWindows XP SP2, Windows Vista







Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace

Note


This namespace is available only when developing for Windows.

Provides base classes that represent the creation and writing of intermediate content for game asset types processed by the XNA Framework Content Pipeline.

Classes

Name	Description
 ContentTypeSerializer	Provides a generic implementation of ContentTypeSerializer methods and properties for serializing and deserializing a specific managed type.
 ContentTypeSerializer	Provides methods for serializing and deserializing a specific managed type.
 ContentTypeSerializerAttribute	Identifies type serializer components.
 IntermediateReader	Provides an implementation of many of the methods of IntermediateSerializer . Deserializes and tracks state for shared resources and external references.
 IntermediateSerializer	Provides methods for reading and writing XNA intermediate XML format.
 IntermediateWriter	Provides an implementation of many of the methods of IntermediateSerializer including serialization and state tracking for shared resources and external references.

Delegates

Name	Description
 ContentTypeSerializer.ChildCallback	Callback delegate for the ScanChildren method.

ContentTypeSerializer Class

Note

This class is available only when developing for Windows.

Provides methods for serializing and deserializing a specific managed type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentTypeSerializer
```

See Also

Reference

[ContentTypeSerializer Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista




ContentTypeSerializer Members

The following tables list the members exposed by the ContentTypeSerializer type.







Public Constructors

Name	Description
 ContentTypeSerializer	Overloaded. Initializes a new instance of the ContentTypeSerializer class.







Public Properties

Name	Description
 CanDeserializeIntoExistingObject	Gets a value indicating whether this component may load data into an existing object or if it must construct a new instance of the object before loading the data.
 TargetType	Gets the type handled by this serializer component.
 XmlTypeName	Gets a short-form XML name for the target type, or null if there is none.

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ObjectsIsEmpty	Queries whether an object contains data to be serialized.
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Deserialize	Deserializes an object from intermediate XML format.
 Finalize	(Inherited from Object .)
 Initialize	Retrieves and caches any nested type serializers and allows reflection over the target data type.
 MemberwiseClone	(Inherited from Object .)
 ScanChildren	Examines the children of the specified object, passing each to a callback delegate.
 Serialize	Serializes an object to intermediate XML format.

See Also

Reference

[ContentTypeSerializer Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer Constructor

Initializes a new instance of the [ContentTypeSerializer](#) class.

Overload List

Name	Description
ContentTypeSerializer (Type)	Initializes a new instance of the ContentTypeSerializer class for serializing the specified type.
ContentTypeSerializer (Type, String)	Initializes a new instance of the ContentTypeSerializer class for serializing the specified type using the specified XML shortcut name.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer Constructor (Type)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeSerializer](#) class for serializing the specified type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentTypeSerializer (  
    Type targetType  
)
```

Parameters

targetType

The target type.

Exceptions

Exception type	Condition
ArgumentNullException	<i>targetType</i> is null .

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer Constructor (Type, String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeSerializer](#) class for serializing the specified type using the specified XML shortcut name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentTypeSerializer (  
    Type targetType,  
    string xmlTypeName  
)
```

Parameters

targetType

The target type.

xmlTypeName

The XML shortcut name.

Exceptions

Exception type	Condition
ArgumentNullException	<i>targetType</i> is null .

See Also

Reference

[ContentTypeSerializer Class](#)







[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)







Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ObjectIsEmpty	Queries whether an object contains data to be serialized.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Deserialize	Deserializes an object from intermediate XML format.
	Finalize	(Inherited from Object .)
	Initialize	Retrieves and caches any nested type serializers and allows reflection over the target data type.
	MemberwiseClone	(Inherited from Object .)
	ScanChildren	Examines the children of the specified object, passing each to a callback delegate.
	Serialize	Serializes an object to intermediate XML format.

See Also

Reference

[ContentTypeSerializer Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.Deserialize Method

Note

This method is available only when developing for Windows.

Deserializes an object from intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal abstract Object Deserialize (  
    IntermediateReader input,  
    ContentTypeSerializerAttribute format,  
    Object existingInstance  
)
```

Parameters

input

Location of the intermediate XML and various deserialization helpers.

format

Specifies the intermediate source XML format.

existingInstance

The object containing the received data, or **null** if the deserializer should construct a new instance.

Return Value

The resultant object.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.Initialize Method

Note

This method is available only when developing for Windows.

Retrieves and caches any nested type serializers and allows reflection over the target data type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal virtual void Initialize (  
    IntermediateSerializer serializer  
)
```

Parameters

serializer

The content serializer.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer.ObjectIsEmpty Method

Note

This method is available only when developing for Windows.

Queries whether an object contains data to be serialized.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ObjectIsEmpty (  
    Object value  
)
```

Parameters

value

The object to query.

Return Value

true if *value* contains data to be serialized; **false** otherwise.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer.ScanChildren Method

Note

This method is available only when developing for Windows.

Examines the children of the specified object, passing each to a callback delegate.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal virtual void ScanChildren (  
    IntermediateSerializer serializer,  
    ChildCallback callback,  
    Object value  
)
```

Parameters

serializer

The content serializer.

callback

The method to be called for each examined child.

value

The object whose children are being scanned.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.Serialize Method

Note

This method is available only when developing for Windows.

Serializes an object to intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal abstract void Serialize (  
    IntermediateWriter output,  
    Object value,  
    ContentTypeSerializerAttribute format  
)
```

Parameters

output

Specifies the intermediate XML location, and provides various serialization helpers.

value

The object to be serialized.

format

Specifies the content format for this object.

See Also

Reference

[ContentTypeSerializer Class](#)




[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer Properties

Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	Gets a value indicating whether this component may load data into an existing object or if it must it construct a new instance of the object before loading the data.
	TargetType	Gets the type handled by this serializer component.
	XmlTypeName	Gets a short-form XML name for the target type, or null if there is none.

See Also

Reference

[ContentTypeSerializer Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.CanDeserializeIntoExistingObject Property

Note

This property is available only when developing for Windows.

Gets a value indicating whether this component may load data into an existing object or if it must construct a new instance of the object before loading the data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool CanDeserializeIntoExistingObject { get; }
```

Property Value

true if this component can load data into an existing object; **false** if a new instance of the object must be constructed.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.TargetType Property

Note

This property is available only when developing for Windows.

Gets the type handled by this serializer component.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Type TargetType { get; }
```

Property Value

The type handled by this serializer component.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.XmlTypeName Property

Note

This property is available only when developing for Windows.

Gets a short-form XML name for the target type, or **null** if there is none.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string XmlTypeName { get; }
```

Property Value

A short-form XML name for the target type, or **null** if there is none.

See Also

Reference

[ContentTypeSerializer Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer Generic Class

Note

This generic class is available only when developing for Windows.

Provides a generic implementation of [ContentTypeSerializer](#) methods and properties for serializing and deserializing a specific managed type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public abstract class ContentTypeSerializer<T> : ContentTypeSerializer
```

Remarks This is a generic implementation of [ContentTypeSerializer](#) and, therefore, can handle strongly typed content data.

See Also

Reference

[ContentTypeSerializer Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista




ContentTypeSerializer Members

The following tables list the members exposed by the ContentTypeSerializer type.










Public Constructors

	Name	Description
	ContentTypeSerializer	Overloaded. Initializes a new instance of the ContentTypeSerializer class.



Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	(Inherited from ContentTypeSerializer .)
	TargetType	(Inherited from ContentTypeSerializer .)
	XmlTypeName	(Inherited from ContentTypeSerializer .)

Public Methods

	Name	Description
	Deserialize	Overloaded. Deserializes an object from intermediate XML format.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ObjectIsEmpty	Overloaded. Queries whether an object contains data to be serialized.
	ReferenceEquals	(Inherited from Object .)
	ScanChildren	Overloaded. Examines the children of the specified object, passing each to a callback delegate.
	Serialize	Overloaded. Serializes an object to intermediate XML format.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer Constructor

Initializes a new instance of the [ContentTypeSerializer](#) class.

Overload List

Name	Description
ContentTypeSerializer ()	Initializes a new instance of the ContentTypeSerializer class.
ContentTypeSerializer (String)	Initializes a new instance of the ContentTypeSerializer class using the specified XML shortcut name.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer Constructor ()

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeSerializer](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentTypeSerializer ()
```

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer Constructor (String)

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeSerializer](#) class using the specified XML shortcut name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected ContentTypeSerializer (  
    string xmlTypeName  
)
```

Parameters

xmlTypeName

The XML shortcut name.

See Also

Reference

[ContentTypeSerializer Generic Class](#)










[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)



Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer Methods

Public Methods

	Name	Description
	Deserialize	Overloaded. Deserializes an object from intermediate XML format.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ObjectIsEmpty	Overloaded. Queries whether an object contains data to be serialized.
	ReferenceEquals	(Inherited from Object .)
	ScanChildren	Overloaded. Examines the children of the specified object, passing each to a callback delegate.
	Serialize	Overloaded. Serializes an object to intermediate XML format.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.Deserialize Method

Deserializes an object from intermediate XML format.

Overload List

Name	Description
ContentTypeSerializer.Deserialize (IntermediateReader, ContentSerializerAttribute, Object)	Deserializes an object from intermediate XML format.
ContentTypeSerializer.Deserialize (IntermediateReader, ContentSerializerAttribute, T)	Deserializes a strongly typed object from intermediate XML format.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.Deserialize Method (IntermediateReader, ContentSerializerAttribute, Object)

Note

This method is available only when developing for Windows.

Deserializes an object from intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal override Object Deserialize (  
    IntermediateReader input,  
    ContentSerializerAttribute format,  
    Object existingInstance  
)
```

Parameters

input

Location of the intermediate XML and various deserialization helpers.

format

Specifies the intermediate source XML format.

existingInstance

The object containing the received data, or **null** if the deserializer should construct a new instance.

Return Value

The object after deserialization.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.Deserialize Method (IntermediateReader, ContentSerializerAttribute, T)

Note

This method is available only when developing for Windows.

Deserializes a strongly typed object from intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal abstract T Deserialize (  
    IntermediateReader input,  
    ContentSerializerAttribute format,  
    T existingInstance  
)
```

Parameters

input

Location of the intermediate XML and various deserialization helpers.

format

Specifies the intermediate source XML format.

existingInstance

The strongly typed object containing the received data, or **null** if the deserializer should construct a new instance.

Return Value

The object type of the deserialized object.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer.ObjectIsEmpty Method

Queries whether an object contains data to be serialized.

Overload List

Name	Description
ContentTypeSerializer.ObjectIsEmpty (Object)	Queries whether a strongly-typed object contains data to be serialized.
ContentTypeSerializer.ObjectIsEmpty (T)	Queries whether an object contains data to be serialized.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.ObjectIsEmpty Method (Object)

Note

This method is available only when developing for Windows.

Queries whether a strongly-typed object contains data to be serialized.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool ObjectIsEmpty (  
    Object value  
)
```

Parameters

value

The object to query.

Return Value

true if *value* contains data to be serialized; **false** otherwise.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.ObjectIsEmpty Method (T)

Note

This method is available only when developing for Windows.

Queries whether an object contains data to be serialized.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public virtual bool ObjectIsEmpty (  
    T value  
)
```

Parameters

value

The object to query.

Return Value

true if *value* contains data to be serialized; **false** otherwise.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializer.ScanChildren Method

Examines the children of the specified object, passing each to a callback delegate.

Overload List

Name	Description
ContentTypeSerializer.ScanChildren (IntermediateSerializer, ChildCallback, Object)	Examines the children of the specified object, passing each to a callback delegate.
ContentTypeSerializer.ScanChildren (IntermediateSerializer, ChildCallback, T)	Examines the children of the specified object, passing each to a callback delegate.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.ScanChildren Method (IntermediateSerializer, ChildCallback, Object)

Note

This method is available only when developing for Windows.

Examines the children of the specified object, passing each to a callback delegate.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal override void ScanChildren (  
    IntermediateSerializer serializer,  
    ChildCallback callback,  
    Object value  
)
```

Parameters

serializer

The content serializer.

callback

The method to be called for each examined child.

value

The object whose children are being scanned.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.ScanChildren Method (IntermediateSerializer, ChildCallback, T)

Note

This method is available only when developing for Windows.

Examines the children of the specified object, passing each to a callback delegate.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal virtual void ScanChildren (  
    IntermediateSerializer serializer,  
    ChildCallback callback,  
    T value  
)
```

Parameters

serializer

The content serializer.

callback

The method to be called for each examined child.

value

The strongly typed object whose children are being scanned.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.Serialize Method

Serializes an object to intermediate XML format.

Overload List

Name	Description
ContentTypeSerializer.Serialize (IntermediateWriter, Object, ContentSerializerAttribute)	Serializes an object to intermediate XML format.
ContentTypeSerializer.Serialize (IntermediateWriter, T, ContentSerializerAttribute)	Serializes an object to intermediate XML format.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializer.Serialize Method (IntermediateWriter, Object, ContentSerializerAttribute)

Note

This method is available only when developing for Windows.

Serializes an object to intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal override void Serialize (  
    IntermediateWriter output,  
    Object value,  
    ContentSerializerAttribute format  
)
```

Parameters

output

Specifies the intermediate XML location, and provides various serialization helpers.

value

The object to be serialized.

format

Specifies the content format for this object.

See Also

Reference

[ContentTypeSerializer Generic Class](#)

[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer.Serialize Method (IntermediateWriter, T, ContentTypeSerializerAttribute)

Note

This method is available only when developing for Windows.

Serializes an object to intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal abstract void Serialize (  
    IntermediateWriter output,  
    T value,  
    ContentTypeSerializerAttribute format  
)
```

Parameters

output

Specifies the intermediate XML location, and provides various serialization helpers.

value

The strongly typed object to be serialized.

format

Specifies the content format for this object.

See Also

Reference

[ContentTypeSerializer Generic Class](#)




[ContentTypeSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializer Properties

Public Properties

	Name	Description
	CanDeserializeIntoExistingObject	(Inherited from ContentTypeSerializer .)
	TargetType	(Inherited from ContentTypeSerializer .)
	XmlTypeName	(Inherited from ContentTypeSerializer .)

See Also

Reference

[ContentTypeSerializer](#) Generic Class

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate](#) Namespace

ContentTypeSerializer.ChildCallback Delegate

Note

This delegate is available only when developing for Windows.

Callback delegate for the [ScanChildren](#) method.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
protected internal delegate void ContentTypeSerializer.ChildCallback (  
    ContentTypeSerializer typeSerializer,  
    Object value  
)
```

Parameters

typeSerializer

The serializer component used to read or write the child object.

value

The child object currently being scanned.

See Also

Reference

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

ContentTypeSerializerAttribute Class

Note

This class is available only when developing for Windows.

Identifies type serializer components.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[AttributeUsageAttribute(4)]  
public sealed class ContentTypeSerializerAttribute : Attribute
```

RemarksAll custom serializers must decorate their class with this attribute, as well as extend the [ContentTypeSerializer](#) class.

See Also

Reference

[ContentTypeSerializer](#)

[ContentTypeSerializerAttribute Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista


ContentTypeSerializerAttribute Members

The following tables list the members exposed by the ContentTypeSerializerAttribute type.










Public Constructors

	Name	Description
	ContentTypeSerializerAttribute	Initializes a new instance of the ContentTypeSerializerAttribute class.



Public Properties

	Name	Description
	TypeId	(Inherited from Attribute .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeSerializerAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializerAttribute Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of the [ContentTypeSerializerAttribute](#) class.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentTypeSerializerAttribute ()
```

See Also

Reference

[ContentTypeSerializerAttribute Class](#)










[ContentTypeSerializerAttribute Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)



PlatformsWindows XP SP2, Windows Vista

ContentTypeSerializerAttribute Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetCustomAttribute	(Inherited from Attribute .)
	GetCustomAttributes	(Inherited from Attribute .)
	GetType	(Inherited from Object .)
	IsDefaultAttribute	(Inherited from Attribute .)
	IsDefined	(Inherited from Attribute .)
	Match	(Inherited from Attribute .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[ContentTypeSerializerAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

ContentTypeSerializerAttribute Properties

Public Properties

	Name	Description
	TypeId	(Inherited from Attribute .)

See Also

Reference

[ContentTypeSerializerAttribute Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateReader Class

Note

This class is available only when developing for Windows.

Provides an implementation of many of the methods of [IntermediateSerializer](#). Deserializes and tracks state for shared resources and external references.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class IntermediateReader
```

RemarksA new instance of **IntermediateReader** is constructed for each deserialize operation.

See Also

Reference

[IntermediateReader Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista












IntermediateReader Members

The following tables list the members exposed by the IntermediateReader type.



Public Properties

	Name	Description
	Serializer	Gets the parent serializer.
	Xml	Gets the XML input stream.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveToElement	Moves to the specified element if the element name exists.
	ReadExternalReference	Reads an external reference ID and records it for subsequent operations.
	ReadObject	Overloaded. Reads a single object from the input XML stream.
	ReadRawObject	Overloaded. Reads a single object from the input XML stream.
	ReadSharedResource	Reads a shared resource ID and records it for subsequent operations.
	ReadTypeName	Reads and decodes a type descriptor from the XML input stream.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also












Reference

[IntermediateReader Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateReader Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	MoveToElement	Moves to the specified element if the element name exists.
	ReadExternalReference	Reads an external reference ID and records it for subsequent operations.
	ReadObject	Overloaded. Reads a single object from the input XML stream.
	ReadRawObject	Overloaded. Reads a single object from the input XML stream.
	ReadSharedResource	Reads a shared resource ID and records it for subsequent operations.
	ReadTypeName	Reads and decodes a type descriptor from the XML input stream.
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[IntermediateReader Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateReader.MoveToElement Method

Note

This method is available only when developing for Windows.

Moves to the specified element if the element name exists.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool MoveToElement (  
    string elementName  
)
```

Parameters

elementName

The element name.

Return Value

true if the specified element exists and the move was successful; **false** otherwise.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateReader.ReadExternalReference Generic Method

Note

This generic method is available only when developing for Windows.

Reads an external reference ID and records it for subsequent operations.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void ReadExternalReference<T> (  
    ExternalReference<T> existingInstance  
)
```

Type Parameters

T

The type of *existingInstance*.

Parameters

existingInstance

The object receiving the data, or **null** if a new instance of the object should be created.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateReader.ReadObject Method

Reads a single object from the input XML stream.

Overload List

Name	Description
IntermediateReader.ReadObject (ContentSerializerAttribute)	Reads a single object from the input XML stream.
IntermediateReader.ReadObject (ContentSerializerAttribute, ContentTypeSerializer)	Reads a single object from the input XML stream, using the specified type hint.
IntermediateReader.ReadObject (ContentSerializerAttribute, ContentTypeSerializer, T)	Reads a single object from the input XML stream using the specified type hint, optionally specifying an existing instance to receive the data.
IntermediateReader.ReadObject (ContentSerializerAttribute, T)	Reads a single object from the input XML stream, optionally specifying an existing instance to receive the data.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateReader.ReadObject Generic Method (ContentSerializerAttribute)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadObject<T> (  
    ContentSerializerAttribute format  
)
```

Type Parameters

T

The type of object read.

Parameters

format

The format expected by the type serializer.

Return Value

The object.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadObject Generic Method (ContentSerializerAttribute, ContentTypeSerializer)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream, using the specified type hint.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadObject<T> (  
    ContentSerializerAttribute format,  
    ContentTypeSerializer typeSerializer  
)
```

Type Parameters

T

The type of object read.

Parameters

format

The format of the XML.

typeSerializer

The type serializer.

Return Value

The object.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadObject Generic Method (ContentSerializerAttribute, ContentTypeSerializer, T)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream using the specified type hint, optionally specifying an existing instance to receive the data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadObject<T> (  
    ContentSerializerAttribute format,  
    ContentTypeSerializer typeSerializer,  
    T existingInstance  
)
```

Type Parameters

T

The type of object read.

Parameters

format

The format of the XML.

typeSerializer

The type serializer.

existingInstance

The object receiving the data, or **null** if a new instance should be created.

Return Value

The object

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadObject Generic Method (ContentSerializerAttribute, T)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream, optionally specifying an existing instance to receive the data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadObject<T> (  
    ContentSerializerAttribute format,  
    T existingInstance  
)
```

Type Parameters

T

The type of object read.

Parameters

format

The format expected by the type serializer.

existingInstance

The object receiving the data, or **null** if a new instance should be created.

Return Value

The object.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadRawObject Method

Reads a single object from the input XML stream.

Overload List

Name	Description
IntermediateReader.ReadRawObject (ContentSerializerAttribute)	Reads a single object from the input XML stream as an instance of the specified type, optionally specifying an existing instance to receive the data.
IntermediateReader.ReadRawObject (ContentSerializerAttribute, ContentTypeSerializer)	Reads a single object from the input XML stream as an instance of the specified type using the specified type hint.
IntermediateReader.ReadRawObject (ContentSerializerAttribute, ContentTypeSerializer, T)	Reads a single object from the input XML stream as an instance of the specified type using the specified type hint, optionally specifying an existing instance to receive the data.
IntermediateReader.ReadRawObject (ContentSerializerAttribute, T)	Reads a single object from the input XML stream, as an instance of the specified type.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateReader.ReadRawObject Generic Method (ContentSerializerAttribute)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream as an instance of the specified type, optionally specifying an existing instance to receive the data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    ContentSerializerAttribute format  
)
```

Type Parameters

T

The type of object to read.

Parameters

format

The format of the XML.

Return Value

The object.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadRawObject Generic Method (ContentSerializerAttribute, ContentTypeSerializer)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream as an instance of the specified type using the specified type hint.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    ContentSerializerAttribute format,  
    ContentTypeSerializer typeSerializer  
)
```

Type Parameters

T

The type of object to read.

Parameters

format

The format of the XML.

typeSerializer

The type serializer.

Return Value

The object.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadRawObject Generic Method (ContentSerializerAttribute, ContentTypeSerializer, T)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream as an instance of the specified type using the specified type hint, optionally specifying an existing instance to receive the data.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    ContentSerializerAttribute format,  
    ContentTypeSerializer typeSerializer,  
    T existingInstance  
)
```

Type Parameters

T

The type of object to read.

Parameters

format

The format of the XML.

typeSerializer

The type serializer.

existingInstance

The object receiving the data, or **null** if a new instance should be created.

Return Value

The object.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadRawObject Generic Method (ContentSerializerAttribute, T)

Note

This generic method is available only when developing for Windows.

Reads a single object from the input XML stream, as an instance of the specified type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public T ReadRawObject<T> (  
    ContentSerializerAttribute format,  
    T existingInstance  
)
```

Type Parameters

T

The type of object to read.

Parameters

format

The object.

existingInstance

The object receiving the data, or **null** if a new instance should be created.

Return Value

The type of object read.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.ReadSharedResource Generic Method

Note

This generic method is available only when developing for Windows.

Reads a shared resource ID and records it for subsequent operations.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void ReadSharedResource<T> (  
    ContentSerializerAttribute format,  
    Action<T> fixup  
)
```

Type Parameters

T

The type to read.

Parameters

format

The format of the XML.

fixup

The fixup operation to perform.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateReader.ReadTypeName Method

Note

This method is available only when developing for Windows.

Reads and decodes a type descriptor from the XML input stream.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public Type ReadTypeName ()
```

Return Value

The type descriptor.

See Also

Reference

[IntermediateReader Class](#)



[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateReader Properties

Public Properties

	Name	Description
	Serializer	Gets the parent serializer.
	Xml	Gets the XML input stream.

See Also

Reference

[IntermediateReader Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateReader.Serializer Property

Note

This property is available only when developing for Windows.

Gets the parent serializer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IntermediateSerializer Serializer { get; }
```

Property Value

The parent serializer.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateReader.Xml Property

Note

This property is available only when developing for Windows.

Gets the XML input stream.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public XmlReader Xml { get; }
```

Property Value

The XML input stream.

See Also

Reference

[IntermediateReader Class](#)

[IntermediateReader Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateSerializer Class

Note

This class is available only when developing for Windows.

Provides methods for reading and writing XNA intermediate XML format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class IntermediateSerializer
```

See Also

Reference

[IntermediateSerializer Members](#)











[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista



IntermediateSerializer Members

The following tables list the members exposed by the IntermediateSerializer type.

Public Methods

	Name	Description
	 Deserialize	Deserializes an intermediate XML file into a managed object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetTypeSerializer	Retrieves the worker serializer for a specified type.
	ReferenceEquals	(Inherited from Object .)
	 Serialize	Serializes an object into an intermediate XML file.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[IntermediateSerializer Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateSerializer Methods

Public Methods

	Name	Description
 	Deserialize	Deserializes an intermediate XML file into a managed object.
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	GetTypeSerializer	Retrieves the worker serializer for a specified type.
	ReferenceEquals	(Inherited from Object .)
 	Serialize	Serializes an object into an intermediate XML file.
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[IntermediateSerializer Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateSerializer.Deserialize Generic Method

Note

This generic method is available only when developing for Windows.

Deserializes an intermediate XML file into a managed object.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static T Deserialize<T> (  
    XmlReader input,  
    string referenceRelocationPath  
)
```

Type Parameters

T

The type to deserialize.

Parameters

input

Intermediate XML file.

referenceRelocationPath

Final name of the output file used to relative encode external reference filenames.

Return Value

The deserialized type.

See Also

Reference

[IntermediateSerializer Class](#)

[IntermediateSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateSerializer.GetTypeSerializer Method

Note

This method is available only when developing for Windows.

Retrieves the worker serializer for a specified type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ContentTypeSerializer GetTypeSerializer (  
    Type type  
)
```

Parameters

type

The type.

Return Value

The worker serializer

See Also

Reference

[IntermediateSerializer Class](#)

[IntermediateSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateSerializer.Serialize Generic Method

Note

This generic method is available only when developing for Windows.

Serializes an object into an intermediate XML file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public static void Serialize<T> (  
    XmlWriter output,  
    T value,  
    string referenceRelocationPath  
)
```

Type Parameters

T

The type of *value*.

Parameters

output

The output XML stream.

value

The object to be serialized.

referenceRelocationPath

Final name of the output file, used to relative encode external reference filenames.

See Also

Reference

[IntermediateSerializer Class](#)

[IntermediateSerializer Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateWriter Class

Note

This class is available only when developing for Windows.

Provides an implementation of many of the methods of [IntermediateSerializer](#) including serialization and state tracking for shared resources and external references.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public sealed class IntermediateWriter
```

RemarksA new instance of **IntermediateWriter** is constructed for each serialize operation.

See Also

Reference

[IntermediateWriter Members](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista











IntermediateWriter Members

The following tables list the members exposed by the IntermediateWriter type.



Public Properties

	Name	Description
	Serializer	Gets the parent serializer.
	Xml	Gets the XML output stream.

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	WriteExternalReference	Adds an external reference to the output XML, and records the filename to be serialized later.
	WriteObject	Overloaded. Writes a single object to the output XML stream.
	WriteRawObject	Overloaded. Writes a single object to the output XML stream.
	WriteSharedResource	Adds a shared reference to the output XML and records the object to be serialized later.
	WriteTypeName	Writes a managed type descriptor to the XML output stream.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also











Reference

[IntermediateWriter Class](#)



[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateWriter Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)
	WriteExternalReference	Adds an external reference to the output XML, and records the filename to be serialized later.
	WriteObject	Overloaded. Writes a single object to the output XML stream.
	WriteRawObject	Overloaded. Writes a single object to the output XML stream.
	WriteSharedResource	Adds a shared reference to the output XML and records the object to be serialized later.
	WriteTypeName	Writes a managed type descriptor to the XML output stream.

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[IntermediateWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateWriter.WriteExternalReference Generic Method

Note

This generic method is available only when developing for Windows.

Adds an external reference to the output XML, and records the filename to be serialized later.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteExternalReference<T> (  
    ExternalReference<T> value  
)
```

Type Parameters

T

The type of external reference to add.

Parameters

value

The external reference to add.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter.WriteObject Method

Writes a single object to the output XML stream.

Overload List

Name	Description
IntermediateWriter.WriteObject (T, ContentSerializerAttribute)	Writes a single object to the output XML stream.
IntermediateWriter.WriteObject (T, ContentSerializerAttribute, ContentSerializer)	Writes a single object to the output XML stream, using the specified type hint.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateWriter.WriteObject Generic Method (T, ContentSerializerAttribute)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output XML stream.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteObject<T> (  
    T value,  
    ContentSerializerAttribute format  
)
```

Type Parameters

T

The type of value to write.

Parameters

value

The value to write.

format

The format of the XML.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter.WriteObject Generic Method (T, ContentSerializerAttribute, ContentTypeSerializer)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output XML stream, using the specified type hint.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteObject<T> (  
    T value,  
    ContentSerializerAttribute format,  
    ContentTypeSerializer typeSerializer  
)
```

Type Parameters

T

The type of value to write.

Parameters

value

The value to write.

format

The format of the XML.

typeSerializer

The type serializer.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter.WriteRawObject Method

Writes a single object to the output XML stream.

Overload List

Name	Description
IntermediateWriter.WriteRawObject (T, ContentSerializerAttribute)	Writes a single object to the output XML stream using the specified serializer worker.
IntermediateWriter.WriteRawObject (T, ContentSerializerAttribute, ContentTypeSerializer)	Writes a single object to the output XML stream as an instance of the specified type.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateWriter.WriteRawObject Generic Method (T, ContentSerializerAttribute)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output XML stream using the specified serializer worker.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteRawObject<T> (  
    T value,  
    ContentSerializerAttribute format  
)
```

Type Parameters

T

The type of value to write.

Parameters

value

The value to write.

format

The format of the XML.

RemarksThis method does not write any type identifier, so it cannot support **null** or polymorphic values and the reader must specify an identical type while deserializing.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter.WriteRawObject Generic Method (T, ContentSerializerAttribute, ContentTypeSerializer)

Note

This generic method is available only when developing for Windows.

Writes a single object to the output XML stream as an instance of the specified type.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteRawObject<T> (  
    T value,  
    ContentSerializerAttribute format,  
    ContentTypeSerializer typeSerializer  
)
```

Type Parameters

T

The type of value to write.

Parameters

value

The value to write.

format

The format of the XML.

typeSerializer

The type serializer.

Remarks If you specify a base class of the actual object value, only data from this base type will be written. This method does not write any type identifier, so it cannot support **null** or polymorphic values and the reader must specify an identical type while deserializing.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista

IntermediateWriter.WriteSharedResource Generic Method

Note

This generic method is available only when developing for Windows.

Adds a shared reference to the output XML and records the object to be serialized later.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteSharedResource<T> (  
    T value,  
    ContentSerializerAttribute format  
)
```

Type Parameters

T

The type of value to write.

Parameters

value

The value to write.

format

The format of the XML.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter.WriteTypeName Method

Note

This method is available only when developing for Windows.

Writes a managed type descriptor to the XML output stream.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public void WriteTypeName (  
    Type type  
)
```

Parameters

type

The type.

See Also

Reference

[IntermediateWriter Class](#)



[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter Properties

Public Properties

	Name	Description
	Serializer	Gets the parent serializer.
	Xml	Gets the XML output stream.

See Also

Reference

[IntermediateWriter Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

IntermediateWriter.Serializer Property

Note

This property is available only when developing for Windows.

Gets the parent serializer.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public IntermediateSerializer Serializer { get; }
```

Property Value

The parent serializer.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

Platforms Windows XP SP2, Windows Vista

IntermediateWriter.Xml Property

Note

This property is available only when developing for Windows.

Gets the XML output stream.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public XmlWriter Xml { get; }
```

Property Value

The XML output stream.

See Also

Reference

[IntermediateWriter Class](#)

[IntermediateWriter Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Serialization.Intermediate Namespace](#)

PlatformsWindows XP SP2, Windows Vista





Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace

Note

This namespace is available only when developing for Windows.

Provides support for importing and processing game assets into the binary format that is used by the content loader of a game project.

Classes

Name	Description
 BuildContent	Provides methods and properties for importing and processing game assets into a binary format.
 BuildXact	An MSBuild task that provides incremental build capabilities for XACT projects.
 CleanContent	An MSBuild task for deleting all the intermediate and output files that were created by a previous Content Pipeline build operation.
 GetLastOutputs	Provides methods and properties for getting the names of all output content files from the content pipeline's cache file.

BuildContent Class

Note

This class is available only when developing for Windows.

Provides methods and properties for importing and processing game assets into a binary format.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class BuildContent : Task
```

Remarks

BuildContent represents a specific [MSBuild](#) task, exposing the complete functionality of the XNA Framework Content Pipeline. This task takes a list of source asset files and the appropriate assemblies (implementing the importer and processor components) as input. After processing, the intermediate result is compiled into a binary format recognized by the game content loader.

See Also

Reference

[BuildContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista


BuildContent Members

The following tables list the members exposed by the BuildContent type.



















Public Constructors

Name	Description
 BuildContent	Initializes a new instance of BuildContent.



Public Fields

Name	Description
 CancelEventNameFormat	The format specifier for the named event used to cancel the build.







Public Properties

Name	Description
 BuildConfiguration	Gets or sets the content build configuration name.
 BuildEngine	(Inherited from Task .)
 BuildEngine2	(Inherited from Task .)
 CompressContent	Gets or sets the content compression flag.
 HostObject	(Inherited from Task .)
 IntermediateDirectory	Gets or sets the directory for storing temporary build files.
 IntermediateFiles	Gets all file names produced by the build, regardless of any incremental optimizations.
 Log	(Inherited from Task .)
 LoggerRootDirectory	Gets or sets the base reference path used when reporting errors during the content build process.
 OutputContentFiles	Gets all file names produced by the build, regardless of any incremental optimizations.
 OutputDirectory	Gets or sets the output directory for the final build results.
 PipelineAssemblies	Gets or sets the names of assemblies that provide Importer<T> or ContentProcessor<T> components for use by the build.
 PipelineAssemblyDependencies	Gets or sets the dependencies of the pipeline assemblies.
 RebuildAll	Gets or sets the force rebuild flag.
 RebuiltContentFiles	Gets the list of file names modified by an incremental rebuild.
 RootDirectory	Gets or sets the base path for the entire content build process.
 SourceAssets	Gets or sets the source asset files to be built.
 TargetPlatform	Gets or sets the content build target platform.



Protected Properties

Name	Description
 HelpKeywordPrefix	(Inherited from Task .)
 TaskResources	(Inherited from Task .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 Execute	Executes the related build task.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also


Reference

[BuildContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildContent Fields

Public Fields

	Name	Description
	CancelEventNameFormat	The format specifier for the named event used to cancel the build.

See Also

Reference

[BuildContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildContent.CancelEventNameFormat Field

Note

This field is available only when developing for Windows.

The format specifier for the named event used to cancel the build.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public const string CancelEventNameFormat
```

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **BuildContent**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BuildContent ()
```

See Also

Reference

[BuildContent Class](#)







[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

BuildContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	Execute	Executes the related build task.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BuildContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildContent.Execute Method

Note

This method is available only when developing for Windows.

Executes the related build task.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool Execute ()
```

Return Value

true if the task completed successfully; **false** otherwise.

See Also

Reference

[BuildContent Class](#)



















[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

BuildContent Properties

Public Properties

	Name	Description
	BuildConfiguration	Gets or sets the content build configuration name.
	BuildEngine	(Inherited from Task .)
	BuildEngine2	(Inherited from Task .)
	CompressContent	Gets or sets the content compression flag.
	HostObject	(Inherited from Task .)
	IntermediateDirectory	Gets or sets the directory for storing temporary build files.
	IntermediateFiles	Gets all file names produced by the build, regardless of any incremental optimizations.
	Log	(Inherited from Task .)
	LoggerRootDirectory	Gets or sets the base reference path used when reporting errors during the content build process.
	OutputContentFiles	Gets all file names produced by the build, regardless of any incremental optimizations.
	OutputDirectory	Gets or sets the output directory for the final build results.
	PipelineAssemblies	Gets or sets the names of assemblies that provide Importer<T> or ContentProcessor<T> components for use by the build.
	PipelineAssemblyDependencies	Gets or sets the dependencies of the pipeline assemblies.
	RebuildAll	Gets or sets the force rebuild flag.
	RebuiltContentFiles	Gets the list of file names modified by an incremental rebuild.
	RootDirectory	Gets or sets the base path for the entire content build process.
	SourceAssets	Gets or sets the source asset files to be built.
	TargetPlatform	Gets or sets the content build target platform.

Protected Properties

	Name	Description
	HelpKeywordPrefix	(Inherited from Task .)
	TaskResources	(Inherited from Task .)

See Also

Reference

[BuildContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildContent.BuildConfiguration Property

Note

This property is available only when developing for Windows.

Gets or sets the content build configuration name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string BuildConfiguration { get; set; }
```

Property Value

Name of the configuration.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.CompressContent Property

Note

This property is available only when developing for Windows.

Gets or sets the content compression flag.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool CompressContent { get; set; }
```

Property Value

If **true**, all content types that permit compression will be compressed when built (the default). If **false**, no content will be compressed.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.IntermediateDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the directory for storing temporary build files.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string IntermediateDirectory { get; set; }
```

Property Value

Directory containing the intermediate build files.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.IntermediateFiles Property

Note

This property is available only when developing for Windows.

Gets all file names produced by the build, regardless of any incremental optimizations. This list can be used as input for a subsequent pack file generator task.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] IntermediateFiles { get; }
```

Property Value

Array of file names produced by the content build.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.LoggerRootDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the base reference path used when reporting errors during the content build process.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string LoggerRootDirectory { get; set; }
```

Property Value

Current name of the base directory or the value to be set.

Remarks

When building XNA content, all errors are displayed in the Error List Window of Visual Studio. Along with the data that triggered the error, the Error List displays the path and filename where the offending data was encountered.

By default, the path of the offending file that is reported in the Error List is always expressed as relative to the location of the project file that references it.

When set, the LoggerRootDirectory property specifies an explicit base path for error reporting that overrides the default operation. All offending files reported in the Error List will be expressed as relative to the base path set in LoggerRootDirectory.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.OutputContentFiles Property

Note

This property is available only when developing for Windows.

Gets all file names produced by the build, regardless of any incremental optimizations. This list can be used as input for a subsequent pack file generator task.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] OutputContentFiles { get; }
```

Property Value

Array of file names produced by the content build.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.OutputDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the output directory for the final build results.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string OutputDirectory { get; set; }
```

Property Value

Output directory for final build result files

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.PipelineAssemblies Property

Note

This property is available only when developing for Windows.

Gets or sets the names of assemblies that provide [Importer<T>](#) or [ContentProcessor<T>](#) components for use by the build.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public ITaskItem[] PipelineAssemblies { get; set; }
```

Property Value

Current pipeline assemblies or the values to be set.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.PipelineAssemblyDependencies Property

Note

This property is available only when developing for Windows.

Gets or sets the dependencies of the pipeline assemblies.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public ITaskItem[] PipelineAssemblyDependencies { get; set; }
```

Property Value

Array of content build dependencies.

Remarks

Any dependencies must be loaded by the task. In addition, these dependencies won't be searched for [ContentImporter](#) or [ContentProcessor](#) components.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.RebuildAll Property

Note

This property is available only when developing for Windows.

Gets or sets the force rebuild flag.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool RebuildAll { get; set; }
```

Property Value

Current value of the force rebuild flag.

If **true**, all content is rebuilt (even when incremental checks indicate everything is up to date). The default value is **false**.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.RebuiltContentFiles Property

Note

This property is available only when developing for Windows.

Gets the list of file names modified by an incremental rebuild. This list is suitable for passing to a subsequent incremental deploy or reload notification task.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] RebuiltContentFiles { get; }
```

Property Value

Array of file names modified by an incremental rebuild.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.RootDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the base path for the entire content build process.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string RootDirectory { get; set; }
```

Property Value

Base path of the content build process.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.SourceAssets Property

Note

This property is available only when developing for Windows.

Gets or sets the source asset files to be built.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]
public ITaskItem[] SourceAssets { get; set; }
```

Property Value

Current source asset files to be built or the values to be set.

Remarks

This task list can be decorated with the following metadata attributes.

Attribute Name	Description
Name	Final name of the compiled asset. This value is passed to the runtime content loader. If you omit this value, it defaults to the source file name with no extension.
Importer	Importer used to read the source asset. If you omit this value, the importer is deduced from the source file extension.
Processor	Content processor used to convert the asset data. If you omit this value, data passes through the pipeline with no custom processing.

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildContent.TargetPlatform Property

Note

This property is available only when developing for Windows.

Gets or sets the content build target platform.

This should be one of the values of the [TargetPlatform Enumeration](#) enumeration: Windows or Xbox360.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string TargetPlatform { get; set; }
```

Property Value

Target of the content build. For a list of possible values, see [TargetPlatform Enumeration](#).

See Also

Reference

[BuildContent Class](#)

[BuildContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact Class

Note

This class is available only when developing for Windows.

An MSBuild task that provides incremental build capabilities for XACT projects.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class BuildXact : Task
```

See Also

Reference

[BuildXact Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

















BuildXact Members

The following tables list the members exposed by the BuildXact type.


Public Constructors

	Name	Description
	BuildXact	Initializes a new instance of BuildXact.







Public Properties

	Name	Description
	BuildConfiguration	Gets or sets the content build configuration name.
	BuildEngine	(Inherited from Task .)
	BuildEngine2	(Inherited from Task .)
	HostObject	(Inherited from Task .)
	IntermediateDirectory	Gets or sets the directory for storing temporary build files.
	IntermediateFiles	Gets the names of the intermediate files used internally by the build.
	Log	(Inherited from Task .)
	LoggerRootDirectory	Gets or sets the base reference path used when reporting errors during the content build process.
	OutputDirectory	Gets or sets the directory for the final build results.
	OutputXactFiles	Gets the names of all files produced by the build, regardless of any incremental optimizations.
	RebuildAll	Gets or sets the force rebuild flag.
	RebuiltXactFiles	Gets the names of files modified by an incremental rebuild.
	RootDirectory	Gets or sets the base directory for the entire content build process.
	TargetPlatform	Gets or sets the content build target platform.
	XactProjects	Gets or sets the XACT project files to be built.
	XnaFrameworkVersion	Gets or sets XNA framework version, used to determine the xactbld3.exe path.



Protected Properties

	Name	Description
	HelpKeywordPrefix	(Inherited from Task .)
	TaskResources	(Inherited from Task .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	Execute	Executes the BuildXact task.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BuildXact Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildXact Constructor

Note

This constructor is available only when developing for Windows.

Initializes a new instance of **BuildXact**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public BuildXact ()
```

See Also

Reference

[BuildXact Class](#)







[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



Platforms Windows XP SP2, Windows Vista

BuildXact Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	Execute	Executes the BuildXact task.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[BuildXact Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildXact.Execute Method

Note

This method is available only when developing for Windows.

Executes the **BuildXact** task.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool Execute ()
```

Return Value

true to begin execution of the task; otherwise **false**.

See Also

Reference

[BuildXact Class](#)

















[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

BuildXact Properties

Public Properties

	Name	Description
	BuildConfiguration	Gets or sets the content build configuration name.
	BuildEngine	(Inherited from Task .)
	BuildEngine2	(Inherited from Task .)
	HostObject	(Inherited from Task .)
	IntermediateDirectory	Gets or sets the directory for storing temporary build files.
	IntermediateFiles	Gets the names of the intermediate files used internally by the build.
	Log	(Inherited from Task .)
	LoggerRootDirectory	Gets or sets the base reference path used when reporting errors during the content build process.
	OutputDirectory	Gets or sets the directory for the final build results.
	OutputXactFiles	Gets the names of all files produced by the build, regardless of any incremental optimizations.
	RebuildAll	Gets or sets the force rebuild flag.
	RebuiltXactFiles	Gets the names of files modified by an incremental rebuild.
	RootDirectory	Gets or sets the base directory for the entire content build process.
	TargetPlatform	Gets or sets the content build target platform.
	XactProjects	Gets or sets the XACT project files to be built.
	XnaFrameworkVersion	Gets or sets XNA framework version, used to determine the xactbld3.exe path.

Protected Properties

	Name	Description
	HelpKeywordPrefix	(Inherited from Task .)
	TaskResources	(Inherited from Task .)

See Also

Reference

[BuildXact Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

BuildXact.BuildConfiguration Property

Note

This property is available only when developing for Windows.

Gets or sets the content build configuration name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string BuildConfiguration { get; set; }
```

Property Value

Current name of the build configuration or the value to be set.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.IntermediateDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the directory for storing temporary build files.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string IntermediateDirectory { get; set; }
```

Property Value

Current directory for temporary build files or the value to be set.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.IntermediateFiles Property

Note

This property is available only when developing for Windows.

Gets the names of the intermediate files used internally by the build.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] IntermediateFiles { get; }
```

Property Value

Intermediate file names produced by the build.

Remarks

When combined with [OutputXactFiles](#), this property gives the complete list of generated files that should be removed by a Clean build operation.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.LoggerRootDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the base reference path used when reporting errors during the content build process.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string LoggerRootDirectory { get; set; }
```

Property Value

Current name of the base directory or the value to be set.

Remarks

When building XACT content, all errors are displayed in the Error List Window of Visual Studio. Along with the data that triggered the error, the Error List displays the path and filename where the offending data was encountered.

By default, the path of the offending file that is reported in the Error List is always expressed as relative to the location of the project file that references it.

When set, the LoggerRootDirectory property specifies an explicit base path for error reporting that overrides the default operation. All offending files reported in the Error List will be expressed as relative to the base path set in LoggerRootDirectory.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.OutputDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the directory for the final build results.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string OutputDirectory { get; set; }
```

Property Value

Current directory for the final build results or the value to be set.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.OutputXactFiles Property

Note

This property is available only when developing for Windows.

Gets the names of all files produced by the build, regardless of any incremental optimizations.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] OutputXactFiles { get; }
```

Property Value

File names produced by the build or the value to be set.

Remarks

Pass this property value to a subsequent packfile generator task or combine with the [IntermediateFiles](#) property to get a complete list of generated files that should be removed by a Clean build operation.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.RebuildAll Property

Note

This property is available only when developing for Windows.

Gets or sets the force rebuild flag.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public bool RebuildAll { get; set; }
```

Property Value

If **true**, all XACT target files are rebuilt; otherwise only modified files are rebuilt. Default value is **false**.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.RebuiltXactFiles Property

Note

This property is available only when developing for Windows.

Gets the names of files modified by an incremental rebuild.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] RebuiltXactFiles { get; }
```

Property Value

File names modified by an incremental build.

Remarks

Optimize your deploy or reload process by passing the current value of **RebuiltXactFiles** to a subsequent incremental deploy or reload notification task.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.RootDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the base directory for the entire content build process.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string RootDirectory { get; set; }
```

Property Value

Current name of the base directory or the value to be set.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.TargetPlatform Property

Note

This property is available only when developing for Windows.

Gets or sets the content build target platform.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string TargetPlatform { get; set; }
```

Property Value

Current platform or the value to be set. Can be one of the following values: Windows or Xbox360

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.XactProjects Property

Note

This property is available only when developing for Windows.

Gets or sets the XACT project files to be built.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public ITaskItem[] XactProjects { get; set; }
```

Property Value

Current XACT project files to be built or value to be set.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

BuildXact.XnaFrameworkVersion Property

Note

This property is available only when developing for Windows.

Gets or sets XNA framework version, used to determine the xactbld3.exe path.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string XnaFrameworkVersion { get; set; }
```

Property Value

Version of the XNA framework or the value to be set. Currently, 1.0 is the only valid value.

See Also

Reference

[BuildXact Class](#)

[BuildXact Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CleanContent Class

Note

This class is available only when developing for Windows.

An MSBuild task for deleting all the intermediate and output files that were created by a previous Content Pipeline build operation.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class CleanContent : Task
```

See Also

Reference

[CleanContent Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista










CleanContent Members

The following tables list the members exposed by the CleanContent type.



Public Constructors

	Name	Description
	CleanContent	Instantiates a new instance of this MSBuild task for deleting all the intermediate and output files that were created by a previous Content Pipeline build operation.







Public Properties

	Name	Description
	BuildConfiguration	Gets or sets the content build configuration name.
	BuildEngine	(Inherited from Task .)
	BuildEngine2	(Inherited from Task .)
	HostObject	(Inherited from Task .)
	IntermediateDirectory	Gets or sets the directory for storing temporary build files.
	Log	(Inherited from Task .)
	OutputDirectory	Gets or sets the output directory for the final build results.
	RootDirectory	Gets or sets the base path for the entire content build process.
	TargetPlatform	Gets or sets the content build target platform.



Protected Properties

	Name	Description
	HelpKeywordPrefix	(Inherited from Task .)
	TaskResources	(Inherited from Task .)

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	Execute	Removes all intermediate and output files that were created by a previous Content Pipeline build operation.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CleanContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

CleanContent Constructor

Note

This constructor is available only when developing for Windows.

Instantiates a new instance of this MSBuild task for deleting all the intermediate and output files that were created by a previous Content Pipeline build operation.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public CleanContent ()
```

See Also

Reference

[CleanContent Class](#)







[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

CleanContent Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	Execute	Removes all intermediate and output files that were created by a previous Content Pipeline build operation.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[CleanContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

CleanContent.Execute Method

Note

This method is available only when developing for Windows.

Removes all intermediate and output files that were created by a previous Content Pipeline build operation.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool Execute ()
```

Return Value

true if errors were logged; **false** otherwise.

See Also

Reference

[CleanContent Class](#)










[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

CleanContent Properties

Public Properties

	Name	Description
	BuildConfiguration	Gets or sets the content build configuration name.
	BuildEngine	(Inherited from Task .)
	BuildEngine2	(Inherited from Task .)
	HostObject	(Inherited from Task .)
	IntermediateDirectory	Gets or sets the directory for storing temporary build files.
	Log	(Inherited from Task .)
	OutputDirectory	Gets or sets the output directory for the final build results.
	RootDirectory	Gets or sets the base path for the entire content build process.
	TargetPlatform	Gets or sets the content build target platform.

Protected Properties

	Name	Description
	HelpKeywordPrefix	(Inherited from Task .)
	TaskResources	(Inherited from Task .)

See Also

Reference

[CleanContent Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

CleanContent.BuildConfiguration Property

Note

This property is available only when developing for Windows.

Gets or sets the content build configuration name.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string BuildConfiguration { get; set; }
```

Property Value

The content build configuration name.

See Also

Reference

[CleanContent Class](#)

[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CleanContent.IntermediateDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the directory for storing temporary build files.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string IntermediateDirectory { get; set; }
```

Property Value

The directory for storing temporary build files.

See Also

Reference

[CleanContent Class](#)

[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CleanContent.OutputDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the output directory for the final build results.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string OutputDirectory { get; set; }
```

Property Value

The output directory for the final build results.

See Also

Reference

[CleanContent Class](#)

[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CleanContent.RootDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the base path for the entire content build process.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public string RootDirectory { get; set; }
```

Property Value

The base path for the entire content build process.

See Also

Reference

[CleanContent Class](#)

[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

CleanContent.TargetPlatform Property

Note

This property is available only when developing for Windows.

Gets or sets the content build target platform.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string TargetPlatform { get; set; }
```

Property Value

The content build target platform.

See Also

Reference

[CleanContent Class](#)

[CleanContent Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GetLastOutputs Class

Note

This class is available only when developing for Windows.

Provides methods and properties for getting the names of all output content files from the content pipeline's cache file.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public class GetLastOutputs : Task
```

See Also

Reference

[GetLastOutputs Members](#)


[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

Platforms Windows XP SP2, Windows Vista







GetLastOutputs Members

The following tables list the members exposed by the GetLastOutputs type.



Public Constructors

Name	Description
 GetLastOutputs	Creates a new instance of GetLastOutputs.







Public Properties

Name	Description
 BuildEngine	(Inherited from Task .)
 BuildEngine2	(Inherited from Task .)
 HostObject	(Inherited from Task .)
 IntermediateDirectory	Gets or sets the directory containing the cache file to be retrieved.
 Log	(Inherited from Task .)
 OutputContentFiles	Gets the names of the output content files.



Protected Properties

Name	Description
 HelpKeywordPrefix	(Inherited from Task .)
 TaskResources	(Inherited from Task .)

Public Methods

Name	Description
 Equals	(Inherited from Object .)
 Execute	Executes the related task using MSBuild.
 GetHashCode	(Inherited from Object .)
 GetType	(Inherited from Object .)
 ReferenceEquals	(Inherited from Object .)
 ToString	(Inherited from Object .)

Protected Methods

Name	Description
 Finalize	(Inherited from Object .)
 MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GetLastOutputs Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

GetLastOutputs Constructor

Note

This constructor is available only when developing for Windows.

Creates a new instance of **GetLastOutputs**.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public GetLastOutputs ()
```

See Also

Reference

[GetLastOutputs Class](#)







[GetLastOutputs Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

GetLastOutputs Methods

Public Methods

	Name	Description
	Equals	(Inherited from Object .)
	Execute	Executes the related task using MSBuild.
	GetHashCode	(Inherited from Object .)
	GetType	(Inherited from Object .)
	ReferenceEquals	(Inherited from Object .)
	ToString	(Inherited from Object .)

Protected Methods

	Name	Description
	Finalize	(Inherited from Object .)
	MemberwiseClone	(Inherited from Object .)

See Also

Reference

[GetLastOutputs Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

GetLastOutputs.Execute Method

Note

This method is available only when developing for Windows.

Executes the related task using [MSBuild](#).

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
public override bool Execute ()
```

Return Value

true if the task completed successfully; **false** otherwise.

See Also

Reference

[GetLastOutputs Class](#)







[GetLastOutputs Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)



PlatformsWindows XP SP2, Windows Vista

GetLastOutputs Properties

Public Properties

	Name	Description
	BuildEngine	(Inherited from Task .)
	BuildEngine2	(Inherited from Task .)
	HostObject	(Inherited from Task .)
	IntermediateDirectory	Gets or sets the directory containing the cache file to be retrieved.
	Log	(Inherited from Task .)
	OutputContentFiles	Gets the names of the output content files.

Protected Properties

	Name	Description
	HelpKeywordPrefix	(Inherited from Task .)
	TaskResources	(Inherited from Task .)

See Also

Reference

[GetLastOutputs Class](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

GetLastOutputs.IntermediateDirectory Property

Note

This property is available only when developing for Windows.

Gets or sets the directory containing the cache file to be retrieved.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[RequiredAttribute]  
public string IntermediateDirectory { get; set; }
```

Property Value

Path of the retrieved cache file.

See Also

Reference

[GetLastOutputs Class](#)

[GetLastOutputs Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

GetLastOutputs.OutputContentFiles Property

Note

This property is available only when developing for Windows.

Gets the names of the output content files. This information may be out of date if a recent build was not completed. The collection is empty if there were no outputs or no cached information was found.

Namespace: Microsoft.Xna.Framework.Content.Pipeline.Tasks

Assembly: Microsoft.Xna.Framework.Content.Pipeline (in microsoft.xna.framework.content.pipeline.dll)

Syntax

C#

```
[OutputAttribute]  
public ITaskItem[] OutputContentFiles { get; }
```

Property Value

Collection of cache file names.

See Also

Reference

[GetLastOutputs Class](#)

[GetLastOutputs Members](#)

[Microsoft.Xna.Framework.Content.Pipeline.Tasks Namespace](#)

PlatformsWindows XP SP2, Windows Vista

White Papers

This section includes technical articles and white papers about programming for XNA Game Studio. Topics can include code migration, performance tweaking, Visual Studio integration, and a host of other subject areas.

In This Section

[API Migration Guide: Managed DirectX 1.1 to XNA Framework](#)

Facilitates the transition of Managed DirectX (MDX) 1.1 developers moving to the XNA Framework.

API Migration Guide: Managed DirectX 1.1 to XNA Framework

There are a number of differences to note when migrating to the XNA Framework. These changes are intentional, and are based on the following design principles for the XNA Framework:

- Embrace the creator community on Microsoft platforms (Windows and Xbox 360 in the initial release).
- Simplify the process of writing games.
- Provide education solutions for academia.
- Provide a positive user experience for MDX or native DirectX developers.

One major architectural difference is the overall scope of XNA Framework versus MDX. While MDX had a 1-to-1 correspondence with many of the core technologies of DirectX, the XNA Framework focuses on an intersection of core DirectX functionality with other game-development technologies.

Another key difference between MDX 1.1 and the XNA Framework is the way the interfaces are packaged. The basic run-time functionality for the XNA Framework is contained in two assemblies:

- Microsoft.Xna.Framework.dll
- Microsoft.Xna.Framework.Game.dll

It is no longer necessary, for example, to reference a separate assembly to add audio support to a project. However, a higher-level differentiation has been made between execution-time and design-time components. Microsoft.Xna.Framework.dll and Microsoft.Xna.Framework.Game.dll correspond to elements from DirectX APIs that are useful for game run times.

XNA Game Studio also contains an extensible XNA Framework Content Pipeline designed to simplify importing, processing, and building game content. A complete representation of the XNA Framework Content Pipeline is outside the scope of this document, but there are several important MDX 1.1 features that are duplicated by XNA Framework Content Pipeline features.

For some developers, porting to the XNA Framework is a less than ideal solution. For one, the XNA Framework API has a significant number of naming differences from the native DirectX APIs. Developers employing MDX tool chains to build content for native games may be more comfortable with the MDX conventions. Also, the XNA Framework does not have support for run-time processing of meshes. This may make the current XNA Framework unsuitable for tools that require run-time import or modification of mesh data.

Getting Started

There are two ways to port a game from MDX to the XNA Framework:

- Reference XNA Framework class libraries in an existing MDX-based project.
- Create a new Windows with XNA Game Studio game and port functionality piece-by-piece from MDX-based sources.

Creating a new XNA Framework game is the recommended porting path. The XNA Game Studio Windows Game template automatically sets up the required references. You can then migrate code files into the new project.

Application Model

Managed DirectX 1.1 and the MDX 2.0 Beta both have functionality for interaction with WinForms. However, the XNA Framework has been designed for an unprecedented level of platform independence. It cannot rely on WinForms for development on Windows and Xbox 360. Therefore, a common set of functionality has been provided in the form of the Application Model. It is important to note that this is not a one-for-one replacement to WinForms, MFC, or Win32; it is intended as a framework specific to game development on Windows and Xbox 360.

To include the application model in a project, you must add the XNA Framework Application Model class library as a reference. The Application Model assembly is Microsoft.Xna.Framework.Game.dll and it can be referenced in the same manner as the XNA Framework assembly.

WinForms to Game

The Game assembly is also a standard partial replacement for the DirectX Managed Utility Toolkit (DXMUT) library previously available as part of the DirectX SDK sample source code. The Application Model does not cover all facets of DXMUT, but it does provide a broad replacement for its boilerplate features.

The advantage of the XNA Framework is that it is compatible with WinForms applications, yet it does not depend on any WinForms or **System.Drawing** types. For applications that already use WinForms, converting to Game is not necessary unless

you are moving to a cross-platform architecture. Also, using the Game interface will add support for the [GraphicsDeviceManager](#), which simplifies device creation and management.

Graphics

XNA Framework Graphics has received a major API overhaul from MDX 1.1, but like MDX, XNA Framework Graphics uses core Direct3D 9 graphics technology. The key interface changes are in the device, effects, D3DX types, and resources. It is also important to note that there is no distinction now between a D3D and a D3DX class. The XNA Framework abstracts the Graphics API to omit any non-intuitive differentiation in the Graphics namespaces.

Another major difference in XNA Framework Graphics is the differentiation of "in-game" and "content creation" functions. In the native DirectX COM API, for example, there is no interface-level distinction between functions that should be used in-game versus those that should be reserved for content creation and processing. The XNA Framework concentrates only on the in-game portions of the API, so some Direct3D 9 features may appear to be missing. Functionality specific to content generation has been moved to its own API and utility set known as the XNA Framework Content Pipeline.

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.Light	No XNA Framework equivalent
Microsoft.DirectX.Direct3D.Material	
Microsoft.DirectX.Direct3D.VertexFormat	

Perhaps the most radical departure from Direct3D 9 is the removal of fixed-function pipeline features from the XNA Framework. The long-term trend in the real time graphics industry has been to move away from fixed-function graphics to the more flexible programmable model. The Xbox 360 does not have a fixed-function pipeline, so shader-based rendering is essential to creating cross-platform games.

The lack of fixed-function rendering in the XNA Framework requires some level of reorganization to port fixed-function rendering engines to the programmable pipeline. Keep in mind that several fixed-function-only classes have been removed altogether and that you must implement a shader-based solution.

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.GraphicsStream	No XNA Framework equivalent

MDX supported a variety of direct memory access operations through the custom GraphicsStream type. Direct memory access is not a feature of the XNA Framework. Data access has been restricted to specific type-safe classes, generic accessors, and arrays. The new access patterns are not only easier to implement, but in many scenarios they are much faster.

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.SwapChain	No XNA Framework equivalent

Swap chains do not exist in the XNA Framework. This was done to improve cross-platform compatibility in the majority of XNA Game Studio rendering scenarios.

Graphics Device Setup

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.Manager	Microsoft.Xna.Framework.Graphics.GraphicsAdapter

The static MDX Manager class has been largely reworked into a more organized set of interfaces based on the [GraphicsAdapter](#) class. The **GraphicsAdapter** class has an [Adapters](#) property, which is a collection of all the adapters on the current machine.

This code example shows how to check a texture format to ensure that it is valid.

C#

```
// Check whether device 0 supports A8R8G8B8 in hardware.
if ( GraphicsAdapter.Adapters[0].CheckDeviceFormat( DeviceType.Hardware,
    SurfaceFormat.Color, ResourceUsage.None, QueryUsage.None,
    ResourceType.Texture2D, Format.A8R8G8B8 ) )
{
    // A8R8G8B8 is supported in hardware.
}
else
```

```
{
    // A8R8B8G8 is not supported in this hardware.
}
```

Likewise, all of the useful **Manager** functions have a similar mapping to methods available from the MDX Manager class.

Graphics Device

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.Device	Microsoft.Xna.Framework.Graphics.GraphicsDevice

[GraphicsDevice](#) behaves like its MDX 1.1 counterpart, without the fixed-function features of the XNA Framework. Most of the methods have been carried over to the new [GraphicsDevice](#), but some have different parameters or overloads.

MDX 1.1	XNA Framework
Device.SetStreamSource	GraphicsDevice.Vertices[]
Device.SetStreamFrequency	
Device.GetStreamSource	
Device.GetStreamSourceFrequency	

The MDX **SetStreamSource** function was used to enable a vertex buffer on a specific stream with a specific stride. Also, the **SetStreamFrequency** method could be used on Shader Model 3.0-compliant hardware to change a specific vertex stream's frequency, enabling performance-boosting techniques such as hardware instancing.

The new access pattern makes intended functionality more discoverable. The supplied index indicates the stream being used. The [VertexStreamCollection](#) returned from this indexer contains properties for the current vertex buffer, stride size, offset, and frequency of vertices on that stream.

For example, to set an 8-byte stride size of vertex stream 0, you could use this C# code.

C#

```
// Set vertex stride for stream 0 to 8 bytes.
device.Vertices[0].VertexStride = 8;
```

MDX 1.1	XNA Framework
Device.SetTexture	GraphicsDevice.Textures[]
Device.GetTexture	

Similar to streams, textures have been abstracted into their own collection representing the different samplers available on the device. The [Textures](#) property is quite straightforward to use:

C#

```
// Set texture stage 0 to myTexture.
device.Textures[0]= myTexture;
```

MDX 1.1	XNA Framework
Device.SetSamplerState	GraphicsDevice.SamplerStates[]
Device.SamplerState[]	

This functionality was available in MDX 1.1. The only difference is that the **SetSamplerState** method has been removed. The [SamplerStates](#) property is a straightforward way of setting and reading all of the possible sampler states for a given stage. As in MDX 1.1, the usage pattern requires an indexer corresponding to the sampler being accessed.

MDX 1.1	XNA Framework
---------	---------------

Device.SetRenderState	GraphicsDevice.RenderState
Device.RenderState	

The [RenderState](#) property was already available in MDX 1.1, but the superfluous **SetRenderState** method has been removed for [GraphicsDevice](#).

MDX 1.1	XNA Framework
Device.Lights	No XNA Framework equivalent
Device.Material	
Device.Transform	
Device.VertexFormat	
Device.RenderState.Ambient	
Device.RenderState.Lighting	
Device.RenderState.LocalViewer	
Device.RenderState.EmissiveMaterialSource	
Device.RenderState.ShadeMode	
Device.RenderState.SpecularEnable	
Device.RenderState.TweenFactor	

Fixed-function pipeline methods have been removed from the XNA Framework, so many MDX **Device** methods and properties do not have an XNA Framework counterpart.

MDX 1.1	XNA Framework
Device.DrawRectanglePatch	No XNA Framework equivalent
Device.DrawTrianglePatch	
Device.NPatchMode	
Device.RenderState.AdaptiveTessellateW	
Device.RenderState.AdaptiveTessellateX	
Device.RenderState.AdaptiveTessellateY	
Device.RenderState.AdaptiveTessellateZ	
Device.RenderState.MaxTessellationValue	
Device.RenderState.MinTessellationValue	
Device.RenderState.PatchEdgeStyle	
Device.RenderState.PositionDegree	
Device.RenderState.NormalizeNormals	

Due to limited hardware support and lack of widespread adoption, patch-based drawing methods have been removed from the XNA Framework.

MDX 1.1	XNA Framework
---------	---------------

Device.ClipStatus	No XNA Framework equivalent
Device.RenderState.UseWBuffer	
Device.RenderState.UpdateSurface	
Device.RenderState.BeginScene	
Device.RenderState.EndScene	

These properties and methods have been removed from the XNA Framework due to limited hardware support, lack of widespread adoption, and Xbox 360 compatibility. **ResourceUsage.Dynamic** resources should be used for default pool textures that require surface updates. **BeginScene** and **EndScene** calls are not required by XNA Framework and have been omitted.

Formats

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.Format	Microsoft.Xna.Graphics.SurfaceFormat

The MDX **Format** type has been renamed [SurfaceFormat](#) for clarity.

MDX 1.1	XNA Framework
Format.Unknown	SurfaceFormat.Unknown
Format.R8G8B8	SurfaceFormat.Bgr24
Format.A8R8G8B8	SurfaceFormat.Color
Format.X8R8G8B8	SurfaceFormat.Bgr32
Format.R5G6B5	SurfaceFormat.Bgr565
Format.X1R5G5B5	SurfaceFormat.Bgr555
Format.A1R5G5B5	SurfaceFormat.Bgra5551
Format.A4R4G4B4	SurfaceFormat.Bgra4444
Format.R3G3B2	SurfaceFormat.Bgr233
Format.A8	SurfaceFormat.Alpha8
Format.A8R3G3B2	SurfaceFormat.Bgra2338
Format.X4R4G4B4	SurfaceFormat.Bgr444
Format.A2B10G10R10	SurfaceFormat.Rgba1010102
Format.A8B8G8R8	SurfaceFormat.Rgba32
Format.X8B8G8R8	SurfaceFormat.Rgb32
Format.G16R16	SurfaceFormat.Rg32
Format.A2R10G10B10	SurfaceFormat.Bgra1010102

Format.A16B16G16R16	SurfaceFormat.Rgba64
Format.A8P8	SurfaceFormat.PaletteAlpha16
Format.P8	SurfaceFormat.Palette8
Format.L8	SurfaceFormat.Luminance8
Format.A8L8	SurfaceFormat.LuminanceAlpha16
Format.A4L4	SurfaceFormat.LuminanceAlpha8
Format.V8U8	SurfaceFormat.NormalizedByte2
Format.L6V5U5	SurfaceFormat.NormalizedLuminance16
Format.X8L8V8U8	SurfaceFormat.NormalizedLuminance32
Format.Q8W8V8U8	SurfaceFormat.NormalizedByte4
Format.V16U16	SurfaceFormat.NormalizedShort2
Format.A2W10V10U10	SurfaceFormat.NormalizedAlpha1010102
Format.D16Lockable	SurfaceFormat.Depth16Lockable
Format.D32	SurfaceFormat.Depth32
Format.D15S1	SurfaceFormat.Depth15Stencil1
Format.D24S8	SurfaceFormat.Depth24Stencil8
Format.D24X8	SurfaceFormat.Depth24
Format.D24X4S4	SurfaceFormat.Depth24Stencil4
Format.D16	SurfaceFormat.Depth16
Format.L16	SurfaceFormat.Luminance16
Format.D32SingleLockable	SurfaceFormat.Depth32SingleLockable
Format.D24SingleS8	SurfaceFormat.Depth24Stencil8Single
Format.Q16W16V16U16	SurfaceFormat.NormalizedShort4
Format.R16F	SurfaceFormat.HalfSingle
Format.G16R16F	SurfaceFormat.HalfVector2
Format.A16B16G16R16F	SurfaceFormat.HalfVector4
Format.R32F	SurfaceFormat.Single

Format.G32R32F	SurfaceFormat.Vector2
Format.A32B32G32R32F	SurfaceFormat.Vector4
Format.CxV8U8	SurfaceFormat.NormalizedByte2Computed
Format.Multi2Argb8	SurfaceFormat.Multi2Bgra32
Format.Dxt1	SurfaceFormat.Dxt1
Format.Dxt2	SurfaceFormat.Dxt2
Format.Dxt3	SurfaceFormat.Dxt3
Format.Dxt4	SurfaceFormat.Dxt4
Format.Dxt5	SurfaceFormat.Dxt5
Format.Yuy2	SurfaceFormat.VideoUyVy
Format.G8R8G8B8	SurfaceFormat.VideoRgBg
Format.VertexData	No XNA Framework equivalent

The full list is included here, as this is one area where the naming convention changed dramatically. The XNA Framework formats have identical layouts to their MDX counterparts.

Effects

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3DX.Effect	Microsoft.Xna.Framework.Graphics.Effect
Microsoft.DirectX.Direct3DX.EffectCompiler	
Microsoft.DirectX.Direct3DX.BaseEffect	
Microsoft.DirectX.Direct3DX.EffectHandle	Microsoft.Xna.Framework.Graphics.EffectAnnotation Microsoft.Xna.Framework.Graphics.EffectFunction Microsoft.Xna.Framework.Graphics.EffectParameter Microsoft.Xna.Framework.Graphics.EffectPass Microsoft.Xna.Framework.Graphics.EffectTechnique

The [Effect](#) interface has changed significantly between the MDX 1.1 implementation and the XNA Framework version. One minor change is that compiled effects were originally accessed through the **GraphicsStream** type. In the XNA Framework, [CompiledEffect](#) takes the place of the MDX 1.1 **GraphicsStream**.

Another feature of XNA Framework is a complete replacement for the **EffectHandle** interface. The XNA Framework implements type-safe replacements to effect handles, which no longer have to be treated as "blobs" of formless data. Effect handles have been abstracted into the types of data or code they represent. The end result is that many potential pitfalls inherited from the native **Effect** functions are eliminated in the XNA Framework.

The new effect handles do not necessarily entail a major departure from the familiar MDX and native Direct3D 9 usage patterns. Some methods have been moved from **Effect** to its subcomponents where it was logical.

The following code is a simplistic example of how to render using a multipass effect technique.

C#

```

// Load a simple effect.
Effect myEffect = new Effect( device, "MyEffect.fx", CompilerOptions.None, null );

// Set parameters.
myEffect.Parameters["g_mWorldViewProjection"].SetValue<Matrix>( Matrix.Identity );

// Render all passes.
myEffect.CurrentTechnique = myEffect.Techniques["T0"];
myEffect.Begin( EffectStateOptions.Default );
foreach( EffectPass pass in myEffect.CurrentTechnique.Passes )
{
    pass.Begin();
    // Draw primitives here....
    pass.End();
}
myEffect.End();

```

Effects in the XNA Framework behave more like a hierarchy than a loose collection of handles into a compiled effect file. Owner relationships have been established between [Effect](#), [EffectTechnique](#), and [EffectPass](#). It follows then that the [Effect.Techniques](#) property returns a collection of techniques, which in turn define a collection of passes located in [EffectTechnique.Passes](#).

Colors and Packed Vectors

MDX 1.1	XNA Framework
System.Drawing.Color	Microsoft.Xna.Framework.Graphics.Color

[System.Drawing](#) is not an XNA Framework dependency, so the XNA Framework contains its own [Color](#) structure. The XNA Framework [Color](#) structure contains all of the colors represented in [System.Drawing.Color](#).

Sprites

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3DX.Sprite	Microsoft.Xna.Framework.Graphics.SpriteBatch

The D3DX [Sprite](#) interface has been redesigned for the XNA Framework into the [SpriteBatch](#) class. It contains functionality similar to D3DX [Sprite](#), but is designed to be more user friendly to 2D and 3D developers alike.

Some functional changes are in the order of operations and the center of Z-rotation for 2D sprites. Sprites always rotate and scale around their center using [SpriteBatch](#). Therefore, [SpriteBatch](#) transforms are applied per each [SpriteBatch.Draw\(\)](#) call instead of per group of sprites.

Shaders

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.VertexShader	Microsoft.Xna.Framework.Graphics.VertexShader
Microsoft.DirectX.Direct3D.PixelShader	Microsoft.Xna.Framework.Graphics.PixelShader
Microsoft.DirectX.Direct3D.VertexDeclaration	Microsoft.Xna.Framework.Graphics.VertexDeclaration
Microsoft.DirectX.Direct3D.ConstantTable Microsoft.DirectX.Direct3D.ConstantTableDescription	Microsoft.Xna.Framework.Graphics.ShaderConstantTable
Microsoft.DirectX.Direct3D.ConstantDescription	Microsoft.Xna.Framework.Graphics.ShaderConstant
Microsoft.DirectX.Direct3D.Semantic	Microsoft.Xna.Framework.Graphics.ShaderSemantic

The basic XNA Framework shader interfaces remain largely unchanged from their MDX and Direct3D 9 counterparts.

MDX 1.1	XNA Framework
---------	---------------

Microsoft.DirectX.Direct3DX.ShaderLoader	Microsoft.Xna.Framework.Graphics.ShaderCompiler Microsoft.Xna.Framework.Graphics.CompiledShader
---	--

The MDX **ShaderLoader** interface has been replaced by a run-time [ShaderCompiler](#) interface in the XNA Framework. In MDX 1.1, the **ShaderLoader** interface would return a GraphicsStream object that could be used to create a vertex or pixel shader. In the XNA Framework, the [ShaderCompiler](#) returns a type-safe [CompiledShader](#), from which you can retrieve input and output semantics.

Meshes and Animation

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3DX.Mesh	No XNA Framework equivalent
Microsoft.DirectX.Direct3DX.BaseMesh	
Microsoft.DirectX.Direct3DX.SkinMesh	
Microsoft.DirectX.Direct3DX.Frame	
Microsoft.DirectX.Direct3DX.BoneCombination	
Microsoft.DirectX.Direct3DX.BoneInfluences	
Microsoft.DirectX.Direct3DX.AttributeRange	
Microsoft.DirectX.Direct3DX.AttributeWeight	
Microsoft.DirectX.Direct3DX.AllocateHierarchy	
Microsoft.DirectX.Direct3DX.AnimationController	
Microsoft.DirectX.Direct3DX.AnimationRootFrame	

The XNA Framework specifies the type [Model](#), which is the run-time component of mesh data imported and processed by the XNA Framework Content Pipeline. The XNA Game Studio documentation includes more information about Model and importing meshes with the XNA Framework Content Pipeline.

Dynamic run-time D3DX **Mesh** loading is not available in the XNA Framework, so an MDX application using D3DX mesh functions, animation controllers, or x-file mesh loading will require game-specific implementation of this functionality.

Graphics Resources

All graphics resources have undergone a revision to their primary data-access pattern. The **Lock** and **Unlock** methods of all resources have been removed in the XNA Framework and have been replaced with **SetData** / **GetData** semantics. **SetData** and **GetData** behave quite differently from the **Lock** / **Unlock** pattern. Using **Lock** / **Unlock**, a given resource could have had read-only, write-only, or read-write properties during a lock. Those ambiguities have been completely cleared up by establishing discreet read and write operations for a resource. That does not mean that the call will always succeed. However, there shouldn't be any confusion as to whether data is being written or read during a given function call.

The **GetData** / **SetData** functions use generic method parameters to indicate the type of the elements within the array. The type provided to the generic parameter indicates the type of the individual elements in the array.

To use **SetData** or **GetData**, it is assumed that you previously allocated a data buffer for use. This is especially important for **GetData**, which, instead of returning a newly allocated array, fills an array provided as a parameter to **GetData**. You must manage the allocation of managed arrays used to read from and write to resources. Resources backed by arrays can help reduce managed heap allocation overhead, as long as the arrays are not constantly re-created.

Textures and Surfaces

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.BaseTexture	Microsoft.Xna.Framework.Graphics.Texture
Microsoft.DirectX.Direct3D.ImageData	
Microsoft.DirectX.Direct3D.Texture	Microsoft.Xna.Framework.Graphics.Texture2D
Microsoft.DirectX.Direct3D.VolumeTexture	Microsoft.Xna.Framework.Graphics.Texture3D

Microsoft.DirectX.Direct3D.CubeTexture	Microsoft.Xna.Framework.Graphics.TextureCube
---	--

Nearly all of the textures and surfaces have different names than they do in native Direct3D 9 and MDX 1.1. The new names are similar to the naming conventions used in Direct3D 10 and are intended to reduce ambiguity about the dimensions of a texture.

Additionally, the **ImageData** information class and other relevant texture data have been moved to the texture itself to remove ambiguities relating to its usage.

MDX 1.1	XNA Framework
Texture.Lock	Texture2D.SetData
Texture.Unlock	Texture2D.GetData
CubeTexture.Lock	TextureCube.SetData
CubeTexture.Unlock	TextureCube.GetData
VolumeTexture.Lock	Texture3D.SetData
VolumeTexture.Unlock	Texture3D.GetData

Like all resources, textures no longer have **Lock** or **Unlock** methods. Data is written and read by using the **SetData** and **GetData** functions described in the Graphics Resources section. The following code snippet is an example usage of the **SetData** and **GetData** methods with textures.

C#

```
// Create a 256 x 256 image in memory.
uint[] imageData = new uint[256 * 256];

// At this point, fill the array with data....

// Fill the texture with the image data.
Texture2D tex = new Texture2D( device, 256, 256, 1, ResourceUsage.None,
SurfaceFormat.Color, ResourceManagementMode.Automatic );
tex.SetData<uint>( imageData );

// Get the image data from the surface.
uint[] imageDataOutput = new uint[256 * 256];
tex.GetData<uint>( imageDataOutput );
```

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3DX.TextureLoader.FromFile	Texture2D.FromFile
Microsoft.DirectX.Direct3DX.TextureLoader.FromCubeFile	TextureCube.FromFile
Microsoft.DirectX.Direct3DX.TextureLoader.FromVolumeFile	Texture3D.FromFile

The XNA Framework does not contain a dedicated static texture loading interface separate from the texture classes themselves. Instead, the static **FromFile** methods are part of each texture type to simplify loading texture resources directly from an image file.

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.Device.GetRendertargetData	Microsoft.Xna.Framework.GraphicsDevice.GetRenderTarget/a>

No MDX equivalent	Microsoft.Xna.Framework.Graphics.RenderTarget Microsoft.Xna.Framework.Graphics.RenderTarget2D Microsoft.Xna.Framework.Graphics.RenderTargetCube
-------------------	---

Render target data is now handled through a strongly-typed object called **RenderTarget**. The **RenderTarget2D.GetTexture** function creates a texture with the render target data. This has made the **Usage.RenderTarget** type obsolete, and it has been omitted in the XNA Framework.

Buffers

MDX 1.1	XNA Framework
Microsoft.DirectX.Direct3D.VertexBuffer	Microsoft.Xna.Framework.Graphics.VertexBuffer
Microsoft.DirectX.Direct3D.IndexBuffer	Microsoft.Xna.Framework.Graphics.IndexBuffer

XNA Framework Index buffers have remained largely unchanged from their MDX counterparts. With fixed-function methods removed, there is no longer a need to provide a vertex buffer's Flexible Vertex Format (FVF) when creating the buffer.

MDX 1.1	XNA Framework
VertexBuffer.Lock	VertexBuffer.SetData
VertexBuffer.Unlock	VertexBuffer.GetData
IndexBuffer.Lock	IndexBuffer.SetData
IndexBuffer.Unlock	IndexBuffer.GetData

Like all graphics resources, the **Lock** and **Unlock** methods have been replaced by dedicated **GetData** and **SetData** calls for reading and writing data respectively.

The following code shows an example usage of the new methods to write and read from a vertex buffer.

C#

```
// Create a set of position vertices.
Vector3[] positionVerts = new Vector3[3];
positionVerts[0] = new Vector3( 0.0f, 0.0f, 0.0f );
positionVerts[1] = new Vector3( 1.0f, 0.0f, 0.0f );
positionVerts[2] = new Vector3( 1.0f, 1.0f, 0.0f );

// Create a vertex buffer and set the data to the position data
// in the positionVerts array.
VertexBuffer vb = new VertexBuffer( device, typeof( Vector3 ), 3,
    ResourceUsage.None, ResourcePool.Managed );
vb.SetData<Vector3>( positionVerts );

// Allocate a new array and fill it with data using GetData.
Vector3[] positionVertsOutput = new Vector3[3];
vb.GetData<Vector3>( positionVertsOutput );
```

MDX 1.1	XNA Framework
VertexBuffer.Description	VertexBuffer

Relevant [VertexBuffer](#) description information has been moved to the [VertexBuffer](#) class rather than a separate **Description** property.

Math

Nearly all of the basic math types familiar to the native D3DX libraries and MDX have been replicated in the XNA Framework. There have been numerous additions to the XNA Framework math types and methods to reduce the amount of boilerplate math code that you have to rewrite for each new game. The biggest breaking change has been the removal of left-handed

coordinate-system math functions in favor of unifying on a right-handed coordinate system. This simplifies the Math API and omits functionality that causes a large amount of confusion.

Vector Math

MDX 1.1	XNA Framework
Microsoft.DirectX.Vector2	Microsoft.Xna.Framework.Vector2
Microsoft.DirectX.Vector3	Microsoft.Xna.Framework.Vector3
Microsoft.DirectX.Vector4	Microsoft.Xna.Framework.Vector4

Vector math is mostly unchanged from its MDX 1.1 implementation. However, several uncommon methods from MDX 1.1 are not available in XNA Framework.

Matrix Math

MDX 1.1	XNA Framework
Microsoft.DirectX.Matrix	Microsoft.Xna.Framework.Matrix

There are several public and static name changes in the new matrix structure. The most common change is that the static **Matrix** methods that create new matrices have been prefixed with the word "**Create**." Also, matrices have a new set of properties that allow rapid access to directional vectors from an orientation matrix.

MDX 1.1	XNA Framework
Matrix.LookAtLH	No XNA Framework equivalent
Matrix.PerspectiveLH	
Matrix.PerspectiveFovLH	
Matrix.PerspectiveOffCenterLH	
Matrix.OrthoLH	
Matrix.OrthoOffCenterLH	
Matrix.LookAtRH	Matrix.CreateLookAt
Matrix.PerspectiveRH	Matrix.CreatePerspective
Matrix.PerspectiveFovRH	Matrix.CreatePerspectiveFieldOfView
Matrix.PerspectiveOffCenterRH	Matrix.CreatePerspectiveOffCenter
Matrix.OrthoRH	Matrix.CreateOrthographic
Matrix.OrthoOffCenterRH	Matrix.CreateOrthographicOffCenter

Right-handed coordinate systems are the default coordinate systems in XNA Framework matrix math. The left-handed matrix methods are not available in the XNA Framework, but you can still create your own left-handed view matrices. Ideally, all new development should be in right-handed coordinates to be compatible with documentation, XNA Framework Content Pipeline components, and XNA Framework drop-in components.

MDX 1.1	XNA Framework
Matrix.TransformCoordinate	Vector3.Transform
Matrix.LookAtRH	Matrix.CreateLookAt

Some matrix method names have been altered to be more consistent with the rest of the XNA Framework.

MDX 1.1	XNA Framework
Matrix.AfflineTransformation	No XNA Framework equivalent
Matrix.AfflineTransformation2D	

Many matrix methods were superfluous or simply elaborate overloads for functionality that can be achieved by multiplying matrices. These were removed to simplify the Math API and clarify math-intensive code.

Quaternion Math

MDX 1.1	XNA Framework
Microsoft.DirectX.Quaternion	Microsoft.Xna.Framework.Quaternion

Quaternion math is unchanged from its MDX 1.1 implementation.

Audio

Audio in XNA Framework is based on the more advanced XACT tools and functions available in the DirectX SDK. In terms of both interface and functionality, XACT is very different from the DirectSound API found in MDX. XACT is both a playback and an authoring system for game audio. It consists of two main components: the XACT API, and the Microsoft Cross-Platform Audio Creation Tool (also called XACT).

XACT is more than a replacement for DirectSound. The run-time XACT functions are easier to use than their DirectSound counterparts. XACT manages the minute details of streaming, threading, and playing individual sounds. The specifics of the audio processing are "baked in" at content creation-time, so the loading and playback of audio can be as efficient as possible.

Using the XACT audio system is a major improvement over DirectSound because it enables cross-platform content creation for audio on Xbox 360 and Windows PC. The tool used to create audio resources for XNA Game Studio (XACT) is the same tool that is used in the Xbox 360 SDK and the DirectX SDK.

The Microsoft Cross-Platform Audio Creation Tool is documented in the DirectX SDK. For more information, see <http://msdn.microsoft.com/directx/sdk/>.

Audio Playback

After you have built an audio project, audio playback is a simple matter as far as coding goes. XACT organizes audio data into a global settings file, banks of wave data, and banks of sounds. Sound banks map audio "Cues" to sounds. Cues have a "play" function that works as you might expect. The following code snippet shows how to play a cue in just a few lines of XNA Framework code.

C#

```
// Load the files built by XACT.
AudioEngine engine = new AudioEngine( "MyAudioSettings.xgs" );
WaveBank wb = new WaveBank( engine, "MyWaveBank.xwb" );
SoundBank sb = new SoundBank( engine, "MySoundBank.xsb" );

// Play the "Noisy" audio cue.
sb.GetCue( "Noisy" ).Play();
```

Input

XNA Framework input is designed around immediate state rather than buffered input events. This design fundamentally changes how input is handled between MDX and XNA Framework applications. Immediate input state polling is cleaner and more flexible than input events, but it requires some re-implementation to work in an existing game.

The key to using state-based input is a simple concept: in-game actions that are triggered on changes in controller inputs require that the game store a previous controller state. For example, to "know" that a button has been pressed during a given frame, the game would have to store the fact that the button was in the released state during the previous frame.

Mouse and Keyboard

In MDX 1.1, you can obtain mouse and keyboard input from a variety of sources. DirectInput has its own mouse and keyboard interfaces supporting both immediate and buffered inputs. It is also typical to use WinForms events or even wndproc (message pump) overrides for input handling.

MDX 1.1	XNA Framework
---------	---------------

No MDX equivalent	Microsoft.Xna.Framework.Input.Mouse
No MDX equivalent	Microsoft.Xna.Framework.Input.Keyboard

In the XNA Framework, mouse and keyboard inputs have been standardized into two interfaces. Unlike MDX or WinForms solutions, there is no setup whatsoever to be done with keyboard or mouse inputs. The following code example shows how to get inputs from a mouse and a keyboard.

C#

```

MouseState mouse = Mouse.GetState();
KeyboardState keyboard = Keyboard.GetState();

if ( mouse.LeftButton == ButtonState.Pressed )
{
    // The left mouse button is down!
}
if ( keyboard.IsKeyDown( Keys.Space ) )
{
    // The SPACEBAR is down!
}

```

Xbox 360 Controller

You can access the Xbox 360 Controller with DirectInput. However, it is highly recommended that XNA Framework-based games take advantage of the classes provided in the namespace [Microsoft.Xna.Framework.Input](#).

MDX 1.1	XNA Framework
No MDX equivalent	Microsoft.Xna.Framework.Input.GamePad

The Xbox 360 Controller for Windows uses the new XInput API, which is streamlined compared to DirectInput. It supports only state-based input. Unlike DirectInput, it does not automatically buffer input events. This means that you must poll input and update game state accordingly.

This code shows how to poll an Xbox 360 Controller for input in just a few lines of code.

C#

```

// Get the controller state using the static GetState method.
GamePadState state = GamePad.GetState( PlayerIndex.One );

// Now check whether the controller is connected.
if ( state.IsConnected )
{
    // Check to see whether Button A is down.
    if ( state.Buttons.A == ButtonState.Pressed )
    {
        // Button A is pressed!
    }
}

```

As was the case for [Keyboard](#) and [Mouse](#), there is no setup required. The first time [GetState](#) is called, the controller is set up and ready for use.

The Xbox 360 Controller also supports vibration. Unlike the complex setup required by DirectInput Force Feedback, vibration can be started or stopped in a single line of XNA Framework code!

C#

```

// Turn on maximum vibration.
GamePad.SetVibration( PlayerIndex.One, 1.0f, 1.0f );

// Turn off vibration.
GamePad.SetVibration( PlayerIndex.One, 0f, 0f );

```

DirectInput Devices

DirectInput devices such as flight sticks, wheels, and older game pads are not supported by the XNA Framework. Such devices are not compatible with the Xbox 360, and thus make poor candidates for cross-platform game support.

For games that require DirectInput device support, the MDX 1.1 DirectInput assembly is still the required means of accessing these controllers. DirectInput does not work across platforms and is not recommended for most XNA Framework projects.

Summary

While the XNA Framework may appear superficially similar to Managed DirectX or even native DirectX, the differences go beneath the surface. Some game ports may be fairly simple, requiring mostly naming changes. Other games, particularly those relying on D3DX animation, complex DirectSound processing, or deeply integrated fixed-function pipeline features may require significant modification to be functional in XNA Game Studio.

Resources

XNA Creators Club Online

<http://creators.xna.com/>

XNA Game Studio Portal

<http://www.microsoft.com/xna/>

DirectX Developer Portal

<http://msdn.microsoft.com/directx/>

XNA Game Studio Team Blog

<http://blogs.msdn.com/xna/>

Microsoft Cross-Platform Audio Creation Tool (XACT)

<http://msdn2.microsoft.com/en-us/library/bb174772.aspx>

XNA Creators Club Online Web Site

Are you ready to take your game development to the next level? Come join the community of game creators at the XNA Creators Club Online Web site. Learn, share, and explore with fellow XNA Creators Club members. Just point your browser to the following address:

<http://creators.xna.com>

What's Inside

As a member of the XNA Creators Club, you can download:

- Educational samples and tutorials
- Technical articles
- Starter kits
- Minigames
- Utilities
- Partner offers

Content is updated regularly, so there is always something new.

Community Forums

Signing up on the XNA Creators Club Online Web site gives you access to the XNA Creators Club Online Forums, where you can interact with other creators. Find the answers to your questions, and share your knowledge with others.

Ready to Join?

The community is waiting for you at the XNA Creators Club Online Web site. Joining is free and easy, so sign up today.

[Visit the XNA Creators Club Online Web Site.](http://creators.xna.com)

Copyright Notice

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, XNA, Xbox, Xbox 360, Xbox LIVE, Zune, Direct3D, DirectMusic, DirectSound, DirectX, Visual C#, Visual C++, Visual Studio .NET, Win32, Windows, and Windows Vista are either registered trademarks or trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.