[MS-SLXV-2010]:
Silverlight XAML Vocabulary Specification 2010

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Table of Contents

Introduction

Xaml is defined in the Xaml Object Mapping Specification (referred to as [MS-XAML] for short). The [MS-XAML] specification enables each application that uses Xaml to define its own vocabulary. Vocabularies are formally specified using the Xaml Schema Information Set, a data model defined by [MS-XAML].

This specification defines the Xaml Schema Information Set information items for the Silverlight Xaml Vocabulary. The information items in the Xaml Schema Information Set presented in this specification can be used in conjunction with [MS-XAML] to determine whether any particular Xaml instance or XML document is valid Silverlight Xaml.

Specification Conventions

The [MS-XAML] specification only defines a data model for the Xaml Schema Information Set. It does not prescribe the representation of the information items that constitute a schema. While [MS-XAML] does introduce a notation with which it defines its intrinsic information items, this specification does not use that notation. Instead, a more compact representation is used to minimize redundancy.

The WPF Xaml Vocabulary contains several thousand information items, many of which share much in common. The following sections describe the conventions used in this specification, which exploit this commonality to reduce the volume of text required to describe each information item, and to make it easier to see each item's distinguishing features.

Xaml Types

The [MS-XAML] specification defines a XamlType Information Item. Throughout this specification, when a XamlType Information Item is presented, only those values that do not match the default value are listed. The following table defines the default values for a XamlType Information Item.

Property	Default Value
[is default constructible]	True
[is nullable]	True
[text syntax]	Null
[dictionary key property]	Null
[name property]	Null
[xml lang property]	Null
[trim surrounding whitespace]	False
[whitespace significant collection]	False
[is list]	False
[is dictionary]	False
[members]	Empty set
[content property]	Null
[allowed types]	Empty set
[allowed key types]	Empty set
[is xdata]	False
[is name scope]	False
[constructors]	Empty set
[return value type]	Null

This specification denotes non-default values for the information set items with rows that begin with the relevant property name in square brackets. Each Xaml Type definition in this specification includes some non-normative information, in order to aid understanding of the type. In order to distinguish them, these rows begin with a name in parentheses.

There is a (usage) row that illustrates the way in which the type is used in XML. If the type cannot contain content, a self-closing tag will be displayed, as the following example shows:

(usage)	<DependencyObject />

Some types offer a choice of content. The following example indicates that the three fixed string values shown are acceptable as content:

(usage)	OnLastWindowClose | OnMainWindowClose | OnExplicitShutdown

Elements that may contain string content look like the following example. Note that there are typically constraints on which strings are valid. The purpose of the (usage) row is only to provide an approximate indication of content, rather than a formal description.

(usage)	<PropertyPath> string </PropertyPath>

Types that can contain an instance of some other type show the name of the type in the usage, as shown in the following example:

(usage)	<BeginStoryboard> Storyboard </BeginStoryboard>

Types that can contain multiple instances of some other type indicate this with an asterisk, as shown in the following example:

(usage)	<XmlNamespaceMappingCollection> XmlNamespaceMapping* </XmlNamespaceMappingCollection>

Some types cannot be used directly in a Xaml document. For example, a type might be used as the [value type] of a member, but may be marked as [is default constructible] False, and have no [text syntax]. A Xaml document might contain types that list such a type in their [types assignable to] property, but the type itself cannot be used directly. (This corresponds to the idea of an abstract type in some object-oriented programming systems.) Such types have "None" in their Usage row.

Type definitions also include (description) rows, both for the type and also for the members of that type. This is also non-normative.

The (used by) row is also non-normative. Each (used by) row provides a list of types that use this type in some way. For example, it lists types that use this type as the [value type] of a member. The (used by) row is provided to make it easier to see how a type is used, and it does not translate into a property in the Xaml Schema Information Set.

Some types that are used as the [value type] of a member, but which are not typically used directly as objects have a (types assignable from) row. This is non-normative, and is provided as a guide to the role of the type. Note that this list is necessarily not complete, because the set of types from which a type is assignable is not closed: anyone is free to define a new Xaml vocabulary which defines types that are assignable to types in this specification.

Xaml Type Order

Xaml types in this specification are ordered in an alphabetical way. The WPF Xaml Vocabulary uses the [types assignable to] property in a way that corresponds to inheritance in object-oriented programming. On the row above each type name is a list of 'Base' types. On the row below each type name are types which directly 'inherit' from that type.

The following example shows the XamlType Information Items for the Fruit, Apple, and Banana types.

Fruit	
Fruit	
AppleBanana	
(usage)	Value
...	...
property N	Value

Fruit> **Apple**	
Apple	
(usage)	Value
...	...
property N	Value

Fruit> Banana	
Banana (4)	
(usage)	Value
...	...
property N	Value

Since this specification models typical object-oriented inheritance, a 'derived' type inherits all members from a 'base' type. This is not made explicit. For each type, only additional members are listed. The [MS-XAML] specification does not require this inheritance-like style. In the Xaml Schema Information Set data model, each type lists its members exhaustively. Therefore, the correct interpretation of a type definition in this specification is that the corresponding XamlType Information Item's [members] property should include not just the listed members, but also all of the [members] of each type listed in its [types assignable to] property.

The "Banana" type, in the example above, has "(4)" listed after it to indicate that this type was introduced in this XAML Vocabulary's version 4 release. All types or members without a version number after it, were released in versions previous to that.

XamlMember Information Items have numerous properties, and in this specification, members are more similar than they are different. So a notation is used to minimize redundancy. Some XamlMember Information Item properties may be omitted. Unless specified otherwise, the default values described in the following table apply.

Property	Default Value
[text syntax]	Null
[is read only]	False
[is static]	False
[is attachable]	False
[target type]	Null
[allowed location]	Any
[is event]	False
[is directive]	False

Members are not defined in distinct sections of this specification - they are listed inside their defining type following a row named (properties). This means that the [owner type] member defined by [MS-XAML] is never specified explicitly in this specification. The [owner type] is always the type in which the member definition appears. Likewise, the [members] property of the defining type is never explicitly defined - it always contains all of the members listed for that type. The [name] and [value type] are specified on the first line of the property description. This line may be followed by non-default values for other properties. The following example shows the XamlType Information Item for the Satsuma type, which defines a member named SegmentCount of type Int32.

Fruit> Satsuma	
Satsuma	
(usage)	<Satsuma />
(description)	Specifies a small, orange citrus fruit.
(properties)	
SegmentCount	Int32
(description)	The number of segments in this satsuma.

If all of the XamlMember Information Item properties had been listed in full for this property, it would look like the following table.

Property	Value
[name]	SegmentCount
[owner type]	Satsuma
[value type]	Int32

Property	Value
[text syntax]	Null
[is read only]	False
[is static]	False
[is attachable]	False
[target type]	Null
[allowed location]	Any
[is event]	False

As with the type-level (description), the per-member (description) entries in this specification are non-normative.

XamlType Information Item descriptions in this document may contain up to three additional member categories: attachable members, event members, and static members. These three member categories have slightly different defaults, and are grouped separately in the type definitions for clarity. The conventions for these member categories are defined in the following sections.

Xaml Members where

A type that defines members whose [is attachable] property is True will list them in a section that begins with "(attachable properties)". The following FruitBowl type example defines a Children member for which the normal defaults apply. This FruitBowl type also defines an attachable member named ZIndex for which [is attachable] is True. (The other member defaults still apply for ZIndex.)

Object > Bowl(T) > FruitBowl	
FruitBowl	
(usage)	<FruitBowl>Fruit*</FruitBowl>
(description)	A container of fruit.
[content property]	Children
(properties)	
Children	FruitCollection
(description)	The items of fruit
(attachable properties)	
FruitBowl.ZIndex	Int32
(description)	Indicates how deeply buried within the fruit bowl a piece of fruit is.
[target type]	Fruit

The name for an attachable member is specified as *TypeName*.*MemberName*. This is a syntactical convention to make it clear that this is an attachable property, and to illustrate how the property will look in a Xaml document. The [name] property of the XamlMember Information Item will only contain the *MemberName* part (the part after the period).

Xaml Types where

A type whose [is generic] property is Truue will be represented with a (T) notation following the [type name]. "Bowl(Fruit)" would be describing a generic Bowl type, with Fruit as the type argument.

Object > Bowl(T)	
Bowl(T)	
FruitBowl	
(usage)	<Bowl x:TypeArguments="T" >
(description)	A container for different types of objects.
[is collection]	True

Xaml Members where

A type that defines members whose [is event] property is True will list them in a section that begins with "(events)". For members in this section, the default value for [value type] is the x:XamlEvent type defined in Section 5 "Intrinsic Schema Information Items" of [MS-XAML]. (The other defaults still apply.)

Xaml Members where

A type that defines members whose [is static] property is True will list them in a section that begins with "(static properties)".

Constructors

Types with a non-empty [constructors] property use a convention similar to that for members. This example includes a constructor:

BruleeToppingExtension	
(usage)	{BruleeToppingExtension}
(description)	Defines the way in which the topping on a brulée-style dessert is prepared.
[types assignable to]	Brulee MarkupExtension Object
(used by)	FruitBrulee CremeBrulee
[return value type]	BruleeTopping
[constructors]	
(2 parameters)	
thickness	Double

BruleeToppingExtension	
(description)	The thickness of the topping in 1/96th of an inch.
burnFactor	Double
(description)	The extent to which the sugar is burnt: 0 for raw sugar, 1 for carbon.

Unlike members, which are identified by names, a constructor for a type is distinguished only by the number of parameters it has. This type has a single constructor that takes two parameters. This is interpreted as a Constructor Information Item, whose [arguments] contains one entry per parameter. The [arguments] list contains just XamlType Information Items - both the Double type in this case. Note that the type is the only formal part of the constructor argument - the name and description are only provided for informational purposes.

The Silverlight Xaml Schema Information Set

The Silverlight Xaml Schema Information Item is a Xaml Schema Information Item (as defined in Section 3 "Xaml Schema Information Set" of [MS-XAML]). Its properties are defined in the following table.

Property	Value
[target namespace]	"http://schemas.microsoft.com/winfx/2006/xaml/presentation"
[types]	All of the XamlType Information Items defined in the "Silverlight XamlType Information Items" section of this specification.
[assignable types]	All of the XamlType Information Items defined in the "Silverlight XamlType Information Items for Assignable Types" section of this specification.
[directives]	The XamlMember Information Items defined in this specification for which the [is directive] property is True.
[compatible with schemas]	"http://schemas.microsoft.com/client/2007"

A Silverlight Xaml instance MUST be well-formed and valid with respect to this schema, using the rules for 'well-formed' and 'valid' defined in [MS-XAML], except as noted in Silverlight Exceptions to the MS-XAML Specification.

An XML document that is a Silverlight Xaml document MUST yield a Silverlight Xaml instance when the processing rules in Section 6 "Creating a Xaml Information Set from XML" are applied using this schema, except as noted in Silverlight Exceptions to the MS-XAML Specification.

Silverlight Exceptions to the MS-XAML Specification

Valid Silverlight Xaml Documents include the set of XamlTypes defined in this specification. The rules for validity of a Xaml document for Silverlight that will create a Xaml Infoset are more constrained than the combination of the XamlTypes plus the specification material in [MS-XAML]. Also, Silverlight's technique for mapping XML into a Xaml Infoset does not completely conform to Section 6 of [MS-XAML]. The following section documents the variances in rules for validation and Infoset processing of a Silverlight Xaml document.

Important: All material in this section applies specifically to Silverlight 4 Xaml documents.

Directives, XML Namespace

5.3.15.S1. xml:base is valid in a Silverlight Xaml Infoset, but information conveyed by xml:base does not change the base URI for a document.

Intrinsic XamlType Information Items, XAML Namespace

Section 5.2 of [MS-XAML] details many Intrinsic XamlType Information Items. However, a valid Silverlight Xaml Infoset MUST NOT contain any of those intrinsic types other than x:NullExtension.

6.6.2.S1. Valid Silverlight Xaml Documents MUST NOT contain x:NullExtension usages of the form: attribute usage of {x:NullExtension}, or element usage of either <x:Null /> or <x:NullExtension />.

Specifically, for x:NullExtension, the following two bullet points do not apply:

- If schema(xmlObjectElement)[types] contains a type 't' where t[name] matches xmlObjectElement[local name], let objectType be that t.
- Otherwise, if schema(xmlObjectElement)[types] contains a type 't' where t[name] matches the concatenation of xmlObjectElement[local name] and 'Extension' and t[types assignable to] contains the x:MarkupExtension type (5.2.21), let objectType be that t.

Intrinsic XamlMember Information Items, XAML Namespace

Section 5.3 of [MS-XAML] details many Intrinsic XamlMember Information Items from the Xaml and XML Namespaces. From the Xaml Namespace, Silverlight Xaml Infosets only support x:Uid, x:Class, x:ClassModifier, x:FieldModifier, x:Key, and x:Name - they MUST NOT contain any other XamlMembers from the Xaml Namespace.

Markup Compatibility

6.3.S1. Markup compatibility refers to the namespace http://schemas.openxmlformats.org/markup-compatibility/2006. The only valid markup compatibility construct in a Silverlight Xaml Infoset is Ignorable. If used, Ignorable MUST be an attribute usage, and MUST be applied on the root element. Element usage or attribute usage on non-root elements are each invalid.

Dictionary Processing

4.3.1.4.S1. The addition of the bold bullet item below into [MS-XAML] section 4.3.1.4 correctly represents the behavior of Silverlight Xaml Document processing.

•Each Object Node Information Item 'dictItem' in [values] MUST match at least one of the following (and let the first of these rules that matches define keyMemberNode for that dictItem):

- dictItem[member nodes] contains a Member Node Information Item that is the x:Key directive (defined in 5.3.5).
- **dictItem[member nodes] contains a Member Node Information Item that is the x:Name directive or the [name property].**
- dictItem[member nodes] contains a Member Node Information Item whose [member] is dictItem[type][dictionary key property].

4.3.1.4.S2. The string used as the key MUST conform to the XamlName Grammar as defined by [MS-XAML].

Member Node Creation from Content

Silverlight has variances in member node creation from content, in cases where the content is provided as x:String.

6.6.6.S1. XamlTypes that are a collection type, where a type that is x:Object exists in [allowed types], should be processed as string content for a [content property]. Silverlight Xaml Documents are invalid if an x:String found in this location. For example, <ItemsControl>Item1</ItemsControl> is not allowed in a valid Silverlight Xaml Document to set the ItemsControl's property named "Items", even though the XamlType of the [content property] for **ItemsControl** ("Items")is x:Object and is thus assignable from x:String.

Silverlight XamlType Information Items

AlignmentX

x:Object > AlignmentX

(usage)	**Left**| **Center**| **Right**
(description)	Describes how content is positioned horizontally in a container.
(used by)	TileBrush
[is nullable]	False
[text syntax]	AlignmentXSyntax

AlignmentY

x:Object > AlignmentY

(usage)	**Top**| **Center**| **Bottom**
(description)	Describes how content is positioned vertically in a container.
(used by)	TileBrush
[is nullable]	False
[text syntax]	AlignmentYSyntax

Analytics

x:Object > Analytics

(usage)	<Analytics />
(description)	Exposes read-only data about how an application is performing.

Application

x:Object > Application

(usage)	<Application />
(description)	Encapsulates a Silverlight application.
(properties)	
ApplicationLifetimeObjects	IList
(description)	The application extension services that have been registered for this application.
[read only]	True

Release: Tuesday, December 23, 2014

(usage)	<Application />
HasElevatedPermissions (4)	x:Boolean
(description)	A value that indicates whether the application is running outside the browser with elevated permissions.
Resources	ResourceDictionary
(description)	A collection of application-scoped resources, such as styles, templates, and brushes.
RootVisual	UIElement
(description)	The main application UI.
(static properties)	
Current	Application
(description)	The Application object for the current application.
(events)	
CheckAndDownloadUpdateCompleted	Occurs when the application has finished checking for updates in response to a CheckAndDownloadUpdateAsync method call.
Exit	Occurs just before an application shuts down and cannot be canceled.
InstallStateChanged	Occurs when the InstallState property value changes.
Startup	Occurs when an application is started.
UnhandledException	Occurs when an exception that is raised by Silverlight is not handled.

ArcSegment

x:Object > DependencyObject > PathSegment > ArcSegment

(usage)	<ArcSegment />
(description)	Represents an elliptical arc between two points.
(properties)	
IsLargeArc	x:Boolean
(description)	A value that indicates whether the arc should be greater than 180 degrees.
Point	Point
(description)	The endpoint of the elliptical arc.
RotationAngle	x:Double
(description)	The amount (in degrees) by which the ellipse is rotated about the x-axis.

(usage)	<ArcSegment />
Size	Size
(description)	The x- and y-radius of the arc as a Size structure.
SweepDirection	SweepDirection
(description)	A value that specifies whether the arc is drawn in the Clockwise or Counterclockwise direction.

AssemblyPart

x:Object > DependencyObject > AssemblyPart

(usage)	<AssemblyPart />
(description)	An assembly part is an assembly that is to be included in a Silverlight-based application package (.xap).
(used by)	AssemblyPartCollection
(properties)	
Source	x:String
(description)	The Uri that identifies an assembly as an assembly part.

AssemblyPartCollection

x:Object > DependencyObject > PresentationFrameworkCollection(AssemblyPart) > AssemblyPartCollection

(usage)	<AssemblyPartCollection>**AssemblyPart***</AssemblyPartCollection>
(description)	Stores a collection of AssemblyPart objects. Provides collection support for the Parts property.
(used by)	Deployment
[is list]	True
[allowed types]	AssemblyPart

AudioCaptureDeviceCollection (4)

x:Object > DependencyObject > PresentationFrameworkCollection(AudioCaptureDevice) > AudioCaptureDeviceCollection

(usage)	None.
(description)	Specifies a collection of AudioCaptureDevice objects.
[is default constructible]	False

(usage)	None.
[is list]	True
[allowed types]	AudioCaptureDevice

AudioSink (4)

x:Object > AudioSink

(usage)	<AudioSink />
(description)	Exposes the capture graph for audio devices. Derive from this type to receive audio information and to obtain the capture graph through CaptureSource.
(properties)	
CaptureSource	CaptureSource
(description)	A capture source that is associated with this AudioSink.

AutomationProperties

x:Object > AutomationProperties

(usage)	None.
(description)	Provides support for getting or setting the value of instance-level values of automation properties. These property values are set as attached properties (typically in XAML) and supplement or override automation property values from a control's AutomationPeer.
[is default constructible]	False
(attachable properties)	
AutomationProperties.AcceleratorKey	x:String
(description)	The accelerator key string for the element.
[target type]	DependencyObject
AutomationProperties.AccessKey	x:String
(description)	The access key for the element.
[target type]	DependencyObject
AutomationProperties.AutomationId	x:String
(description)	The string that uniquely identifies the element to UI automation.
[target type]	DependencyObject
AutomationProperties.HelpText	x:String

(usage)	None.
(description)	The help text for the element.
[target type]	DependencyObject
AutomationProperties.IsRequiredForForm	x:Boolean
(description)	A value that indicates whether the element is required to be filled out on a form.
[target type]	DependencyObject
AutomationProperties.ItemStatus	x:String
(description)	A description of the status of an item in an element.
[target type]	DependencyObject
AutomationProperties.ItemType	x:String
(description)	A description of the type of the specified element.
[target type]	DependencyObject
AutomationProperties.LabeledBy	UIElement
(description)	The element that contains the text label for the element.
[target type]	DependencyObject
AutomationProperties.Name	x:String
(description)	The automation name of the element.
[target type]	DependencyObject

BackEase

x:Object > DependencyObject > EasingFunctionBase > BackEase, IEasingFunction

(usage)	<BackEase />
(description)	Represents an easing function that retracts the motion of an animation slightly before it begins to animate in the path indicated.
(properties)	
Amplitude	x:Double
(description)	The amplitude of retraction associated with a BackEase animation.

BeginStoryboard

x:Object > DependencyObject > TriggerAction > BeginStoryboard

(usage)	<BeginStoryboard>**Storyboard**</BeginStoryboard>

(usage)	<BeginStoryboard>**Storyboard**</BeginStoryboard>
(description)	A trigger action that begins a Storyboard and distributes its animations to their targeted objects and properties.
[content property]	Storyboard
(properties)	
Storyboard	Storyboard
(description)	The Storyboard that this BeginStoryboard starts.

BezierSegment

x:Object > DependencyObject > PathSegment > BezierSegment

(usage)	<BezierSegment />
(description)	Represents a cubic Bezier curve drawn between two points.
(properties)	
Point1	Point
(description)	The first control point of the curve.
Point2	Point
(description)	The second control point of the curve.
Point3	Point
(description)	The end point of the curve.

Binding

x:Object > x:MarkupExtension > BindingBase > Binding

(usage)	{Binding } | <Binding />
(description)	Defines a binding that connects the properties of binding targets and data sources.
[return value type]	x:Object
[constructors]	
(1 parameter)	
path	x:String
(description)	The initial property path for the source of the binding.
(properties)	

(usage)	{Binding } | <Binding />
BindsDirectlyToSource	x:Boolean
(description)	A value that indicates whether the binding ignores any ICollectionView settings on the data source.
ConverterParameter	x:Object
(description)	A parameter that can be used in the Converter logic.
ElementName	x:String
(description)	The name of the element to use as the binding source object.
Mode	BindingMode
(description)	A value that indicates the direction of the data flow in the binding.
NotifyOnValidationError	x:Boolean
(description)	A value that indicates whether the BindingValidationError event is raised on validation errors.
Path	PropertyPath
(description)	The path to the binding source property.
[text syntax]	PropertyPathSyntax
RelativeSource	RelativeSource
(description)	The binding source by specifying its location relative to the position of the binding target.
Source	x:Object
(description)	The data source for the binding.
UpdateSourceTrigger	UpdateSourceTrigger
(description)	A value that determines the timing of binding source updates for two-way bindings.
ValidatesOnDataErrors (4)	x:Boolean
(description)	A value that indicates whether the binding engine will report validation errors from an IDataErrorInfo implementation on the bound data entity.
ValidatesOnExceptions	x:Boolean
(description)	A value that indicates whether the binding engine will report exception validation errors.
ValidatesOnNotifyDataErrors (4)	x:Boolean
(description)	A value that indicates whether the binding engine will report validation errors from an INotifyDataErrorInfo implementation on the

(usage)	{Binding } | <Binding />
	bound data entity.

BindingBase

x:Object > x:MarkupExtension > BindingBase

Binding	
(usage)	None.
(description)	A base type for the Binding type.
[is default constructible]	False
[return value type]	x:Object
(properties)	
FallbackValue (4)	x:Object
(description)	The value to use when the binding is unable to return a value.
StringFormat (4)	x:String
(description)	A string that specifies how to format the binding if it displays the bound value as a string.
TargetNullValue (4)	x:Object
(description)	The value that is used in the target when the value of the source is null.

BindingMode

x:Object > BindingMode

(usage)	**OneWay**| **OneTime**| **TwoWay**
(description)	Describes how the data propagates in a binding.
(used by)	Binding
[is nullable]	False
[text syntax]	BindingModeSyntax

BitmapCache

x:Object > DependencyObject > CacheMode > BitmapCache

(usage)	<BitmapCache />
(description)	Represents the behavior of caching a visual element or tree of elements as bitmap surfaces. This can yield significant performance improvements for some scenarios.

(usage)	<BitmapCache />
(properties)	
RenderAtScale	x:Double
(description)	The scale at which the object is rendered on the cached bitmap surface. Use this property for cached objects that are scaled to improve performance.

BitmapCreateOptions

x:Object > BitmapCreateOptions

(usage)	**None**| **DelayCreation**| **IgnoreImageCache**
(description)	Specifies initialization options for a bitmap image.
(used by)	BitmapImage
[is nullable]	False
[text syntax]	BitmapCreateOptionsSyntax

BitmapImage

x:Object > DependencyObject > ImageSource > BitmapSource > BitmapImage

(usage)	<BitmapImage />
(description)	Provides the practical object source type for the Source and ImageSource properties.
(properties)	
CreateOptions	BitmapCreateOptions
(description)	The BitmapCreateOptions for a BitmapImage.
UriSource	x:Uri
(description)	The URI of the graphics source file that generated this BitmapImage.
(events)	
DownloadProgress	Occurs when a significant change has occurred in the download progress of the BitmapImage content.
ImageFailed	Occurs when there is an error associated with image retrieval or format.
ImageOpened	Occurs when the image source is downloaded and decoded with no failure. You can use this event to determine the size of an image before rendering it.

BitmapSource

x:Object > DependencyObject > ImageSource > BitmapSource

BitmapImage	
(usage)	None.
(description)	Provides a source object for properties that use a bitmap.
[is default constructible]	False

Block (4)

x:Object > DependencyObject > TextElement > Block

Paragraph	
(usage)	None.
(description)	A base type that provides a base for all block-level content elements.
(used by)	BlockCollection
[is default constructible]	False
(properties)	
TextAlignment	TextAlignment
(description)	The horizontal alignment of the text content.

BlockCollection (4)

x:Object > DependencyObject > TextElementCollection(Block) > BlockCollection

(usage)	None.
(description)	Represents a collection of Block elements.
(used by)	RichTextBox
[is default constructible]	False
[is list]	True
[allowed types]	Block

BlurEffect

x:Object > DependencyObject > Effect > BlurEffect

(usage)	<BlurEffect />
(description)	Represents an effect that you can apply to an object that simulates looking at the object through an out-of-focus lens.
(properties)	
Radius	x:Double

(usage)	<BlurEffect />
(description)	The amount of blurring applied by the BlurEffect.

Bold (4)

x:Object > DependencyObject > TextElement > Inline > Span > Bold

(usage)	<Bold>[**Inline**|**x:String**|**UIElement**]*</Bold>
(description)	Provides an inline-level content element that causes content to appear with a bold font weight.
[content property]	Inlines
[xml lang property]	Language

Border

x:Object > DependencyObject > UIElement > FrameworkElement > Border

(usage)	<Border>**UIElement**</Border>
(description)	Draws a border, background, or both around another object.
[content property]	Child
[name property]	Name
[xml lang property]	Language
(properties)	
Background	Brush
(description)	The Brush that fills the background of the border.
BorderBrush	Brush
(description)	The Brush that is used to create the border.
BorderThickness	Thickness
(description)	The thickness of the border.
Child	UIElement
(description)	The child element to draw the border around.
CornerRadius	CornerRadius
(description)	The radius for the corners of the border.
Padding	Thickness

(usage)	<Border>**UIElement**</Border>
(description)	The distance between the border and its child object.

BounceEase

x:Object > DependencyObject > EasingFunctionBase > BounceEase, IEasingFunction

(usage)	<BounceEase />
(description)	Represents an easing function that creates an animated bouncing effect.
(properties)	
Bounces	x:Int32
(description)	The number of bounces.
Bounciness	x:Double
(description)	A value that specifies how bouncy the bounce animation is. Low values of this property result in bounces with little lose of height between bounces (more bouncy) while high values result in dampened bounces (less bouncy).

Brush

x:Object > DependencyObject > Brush

GradientBrush ImplicitInputBrush SolidColorBrush TileBrush	
(usage)	<Brush> string </Brush>
(description)	Defines objects used to paint graphical objects. Types that derive from Brush describe how the area is painted.
(used by)	Border Control Glyphs Hyperlink Inline Panel PasswordBox RichTextBox Shape TextBlock TextBox TextElement UIElement
[is default constructible]	False
[text syntax]	BrushSyntax
(properties)	
Opacity	x:Double
(description)	The degree of opacity of a Brush.
RelativeTransform	Transform
(description)	The transformation that is applied to the brush using relative coordinates.
Transform	Transform
(description)	The transformation that is applied to the brush.

BrushMappingMode

x:Object > BrushMappingMode

(usage)	**Absolute**| **RelativeToBoundingBox**
(description)	Specifies the coordinate system used by a Brush.
(used by)	GradientBrush
[is nullable]	False
[text syntax]	BrushMappingModeSyntax

Button

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase > Button

(usage)	<Button>**x:Object**</Button>
(description)	Represents a button control.
[content property]	Content
[name property]	Name
[xml lang property]	Language

ButtonBase

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase

Button HyperlinkButton RepeatButton ToggleButton	
(usage)	None.
(description)	Represents the base type for all button controls, such as Button, RepeatButton, and HyperlinkButton.
[is default constructible]	False
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
ClickMode	ClickMode
(description)	When the Click event occurs.
CommandParameter	x:Object

Button HyperlinkButton RepeatButton ToggleButton	
(4)	
(description)	The parameter to pass to the Command property.
IsFocused	x:Boolean
(description)	A value that determines whether the button has focus.
IsMouseOver	x:Boolean
(description)	A value indicating whether the mouse pointer is located over this button control.
IsPressed	x:Boolean
(description)	A value that indicates whether a ButtonBase is currently in a pressed state.
(events)	
Click	Occurs when a Button is clicked.

CacheMode

x:Object > DependencyObject > CacheMode

BitmapCache	
(usage)	<CacheMode> string </CacheMode>
(description)	Represents cached content.
(used by)	UIElement
[is default constructible]	False
[text syntax]	CacheModeSyntax

Canvas

x:Object > DependencyObject > UIElement > FrameworkElement > Panel > Canvas

InkPresenter	
(usage)	<Canvas>UIElement*</Canvas>
(description)	Defines an area within which you can explicitly position child objects by using coordinates that are relative to the area.
[content property]	Children
[name property]	Name
[xml lang property]	Language
(attachable	

InkPresenter	
properties)	
Canvas.Left	x:Double
(description)	The distance between the left side of an object and the left side of its parent Canvas.
[target type]	UIElement
Canvas.Top	x:Double
(description)	The distance between the top of an element and the top of its parent Canvas.
[target type]	UIElement
Canvas.ZIndex	x:Int32
(description)	The z-order rendering behavior of objects in a Canvas.
[target type]	UIElement

CaptureDeviceConfiguration (4)

x:Object > CaptureDeviceConfiguration

(usage)	{x:Static CaptureDeviceConfiguration.StaticPropertyName}
(description)	Represents a helper type for obtaining information about available capture devices (audio or video) and requesting client user permission to access the captures from available devices.
[is default constructible]	False
(static properties)	
AllowedDeviceAccess	x:Boolean
(description)	A value that reports whether a user has previously granted device access based on their stored response to the device access UI prompt.

CaptureSource (4)

x:Object > DependencyObject > CaptureSource

(usage)	<CaptureSource />
(description)	Provides methods that work with specific audio or video captures from the associated capture device.
(used by)	AudioSink VideoSink
(events)	
CaptureFailed	Occurs when a requested capture operation does not generate a captured sample.

(usage)	<CaptureSource />
CaptureImageCompleted	Occurs when an asynchronous image capture request has returned a captured image.

CaptureState (4)

x:Object > CaptureState

(usage)	**Stopped**| **Started**| **Failed**
(description)	Describes the current operation state for device capture and a capture source.
[is nullable]	False
[text syntax]	CaptureStateSyntax

CheckBox

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase > ToggleButton > CheckBox

(usage)	<CheckBox>**x:Object**</CheckBox>
(description)	Represents a control that a user can select (check) or clear (uncheck).
[content property]	Content
[name property]	Name
[xml lang property]	Language

CircleEase

x:Object > DependencyObject > EasingFunctionBase > CircleEase, IEasingFunction

(usage)	<CircleEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using a circular function.

ClickMode

x:Object > ClickMode

(usage)	**Release**| **Press**| **Hover**
(description)	Specifies when the Click event should be raised for a control.
(used by)	ButtonBase
[is nullable]	False
[text syntax]	ClickModeSyntax

ClockState

x:Object > ClockState

(usage)	**Active**| **Filling**| **Stopped**
(description)	Describes the potential states of an animation.
[is nullable]	False
[text syntax]	ClockStateSyntax

CollectionViewSource

x:Object > DependencyObject > CollectionViewSource

(usage)	<CollectionViewSource />
(description)	The XAML proxy of a collection view type.
(properties)	
GroupDescriptions	ObservableCollection(GroupDescription)
(description)	A collection of GroupDescription objects that describe how items in the collection are grouped in the view.
[read only]	True
SortDescriptions	SortDescriptionCollection
(description)	A collection of SortDescription objects that describe how the items in the collection are sorted in the view.
[read only]	True
Source	x:Object
(description)	The collection object from which to create this view.
(events)	
Filter	Provides filtering logic.

Color

x:Object > Color

(usage)	**Black**| **Blue**| **Brown**| **Cyan**| **DarkGray**| **Gray**| **Green**| **LightGray**| **Magenta**...
(description)	Describes a color in terms of alpha, red, green, and blue channels.
(used by)	ColorKeyFrame DrawingAttributes DropShadowEffect GradientStop SolidColorBrush
[is nullable]	False
[text syntax]	ColorSyntax

(usage)	**Black**| **Blue**| **Brown**| **Cyan**| **DarkGray**| **Gray**| **Green**| **LightGray**| **Magenta**...
(properties)	
A	x:Byte
(description)	The sRGB alpha channel value of the color.
B	x:Byte
(description)	The sRGB blue channel value of the color.
G	x:Byte
(description)	The sRGB green channel value of the color.
R	x:Byte
(description)	The sRGB red channel value of the color.

ColorAnimation

x:Object > DependencyObject > Timeline > ColorAnimation

(usage)	<ColorAnimation />
(description)	Animates the value of a Color property between two target values using linear interpolation over a specified Duration.
(properties)	
By	x:Nullable(Color)
(description)	The total amount by which the animation changes its starting value.
EasingFunction	IEasingFunction
(description)	The easing function applied to this animation.
From	x:Nullable(Color)
(description)	The animation's starting value.
To	x:Nullable(Color)
(description)	The animation's ending value.

ColorAnimationUsingKeyFrames

x:Object > DependencyObject > Timeline > ColorAnimationUsingKeyFrames

(usage)	<ColorAnimationUsingKeyFrames>**ColorKeyFrame***</ColorAnimationUsingKeyFrames>
(description)	Animates the value of a Color property along a set of KeyFrames over a specified Duration.
[content	KeyFrames

(usage)	<ColorAnimationUsingKeyFrames>**ColorKeyFrame***</ColorAnimationUsingKeyFrames>
property]	
(properties)	
KeyFrames	ColorKeyFrameCollection
(description)	The collection of ColorKeyFrame objects that define the animation.
[read only]	True

ColorInterpolationMode

x:Object > ColorInterpolationMode

(usage)	**ScRgbLinearInterpolation**| **SRgbLinearInterpolation**
(description)	Determines how the colors in a gradient are interpolated.
(used by)	GradientBrush
[is nullable]	False
[text syntax]	ColorInterpolationModeSyntax

ColorKeyFrame

x:Object > DependencyObject > ColorKeyFrame

DiscreteColorKeyFrame EasingColorKeyFrame LinearColorKeyFrame SplineColorKeyFrame	
(usage)	None.
(description)	Provides a base type for specific animation key-frame techniques that define an animation segment with a Color target value. Derived types each provide a different key-frame interpolation technique for a Color value that is provided for a ColorAnimationUsingKeyFrames animation.
(used by)	ColorKeyFrameCollection
[is default constructible]	False
(properties)	
KeyTime	KeyTime
(description)	The time at which the key frame's target Value should be reached.
Value	Color
(description)	The key frame's target value.

ColorKeyFrameCollection

x:Object > DependencyObject > PresentationFrameworkCollection(ColorKeyFrame) > ColorKeyFrameCollection

(usage)	<ColorKeyFrameCollection>**ColorKeyFrame***</ColorKeyFrameCollection>
(description)	Represents a collection of ColorKeyFrame objects that can be individually accessed by index.
(used by)	ColorAnimationUsingKeyFrames
[is list]	True
[allowed types]	ColorKeyFrame

Colors

x:Object > Colors

(usage)	{x:Static Colors.StaticPropertyName}
(description)	Implements a set of predefined colors.
[is default constructible]	False
(static properties)	
Black	Color
(description)	The system-defined color that has the ARGB value of #FF000000.
Blue	Color
(description)	The system-defined color that has the ARGB value of #FF0000FF.
Brown	Color
(description)	The system-defined color that has the ARGB value of #FFA52A2A.
Cyan	Color
(description)	The system-defined color that has the ARGB value of #FF00FFFF.
DarkGray	Color
(description)	The system-defined color that has the ARGB value of #FFA9A9A9.
Gray	Color
(description)	The system-defined color that has the ARGB value of #FF808080.
Green	Color
(description)	The system-defined color that has the ARGB value of #FF008000.
LightGray	Color

(usage)	{x:Static Colors.StaticPropertyName}
(description)	The system-defined color that has the ARGB value of #FFD3D3D3.
Magenta	Color
(description)	The system-defined color that has the ARGB value of #FFFF00FF.
Orange	Color
(description)	The system-defined color that has the ARGB value of #FFFFA500.
Purple	Color
(description)	The system-defined color that has the ARGB value of #FF800080.
Red	Color
(description)	The system-defined color that has the ARGB value of #FFFF0000.
Transparent	Color
(description)	The system-defined color that has the ARGB value of #00FFFFFF.
White	Color
(description)	The system-defined color that has the ARGB value of #FFFFFFFF.
Yellow	Color
(description)	The system-defined color that has the ARGB value of #FFFFFF00.

ColumnDefinition

x:Object > DependencyObject > ColumnDefinition

(usage)	<ColumnDefinition />
(description)	Defines column-specific properties that apply to Grid objects.
(used by)	ColumnDefinitionCollection
(properties)	
MaxWidth	x:Double
(description)	A value that represents the maximum width of a ColumnDefinition.
MinWidth	x:Double
(description)	A value that represents the minimum width of a ColumnDefinition.
Width	GridLength
(description)	A value that represents the width of a ColumnDefinition.

ColumnDefinitionCollection

x:Object > DependencyObject > PresentationFrameworkCollection(ColumnDefinition) > ColumnDefinitionCollection

(usage)	None.
(description)	Provides access to an ordered, strongly typed collection of ColumnDefinition objects.
(used by)	Grid
[is default constructible]	False
[is list]	True
[allowed types]	ColumnDefinition

ComboBox

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ItemsControl > Selector > ComboBox

(usage)	<ComboBox>**x:Object***</ComboBox>
(description)	Represents a selection control that combines a non-editable text box and a drop-down containing a list box that allows users to select an item from a list.
[content property]	Items
[name property]	Name
[xml lang property]	Language
(properties)	
IsDropDownOpen	x:Boolean
(description)	A value that indicates whether the drop-down portion of the combo box is currently open.
IsSelectionBoxHighlighted	x:Boolean
(description)	A value that indicates whether the SelectionBoxItem is highlighted.
ItemContainerStyle	Style
(description)	The style applied to the container generated for each item in the combo box.
MaxDropDownHeight	x:Double
(description)	The maximum height of the drop-down that lists combo box items.
SelectionBoxItem	x:Object

(usage)	<ComboBox>**x:Object***</ComboBox>
(description)	The item displayed in the selection box.
SelectionBoxItemTemplate	DataTemplate
(description)	The template applied to the selection box content.
(events)	
DropDownClosed	Occurs when the drop-down portion of the combo box closes.
DropDownOpened	Occurs when the drop-down portion of the combo box opens.

ComboBoxItem

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ListBoxItem > ComboBoxItem

(usage)	<ComboBoxItem>**x:Object**</ComboBoxItem>
(description)	Represents a selectable item contained in a ComboBox control.
[content property]	Content
[name property]	Name
[xml lang property]	Language

CompositeTransform (4)

x:Object > DependencyObject > GeneralTransform > Transform > CompositeTransform

(usage)	<CompositeTransform />
(description)	This type lets you apply multiple different transforms to an object.
(properties)	
CenterX	x:Double
(description)	The x-coordinate of the center point for all transforms specified by the CompositeTransform.
CenterY	x:Double
(description)	The y-coordinate of the center point for all transforms specified by the CompositeTransform.
Rotation	x:Double
(description)	The angle, in degrees, of clockwise rotation.
ScaleX	x:Double
(description)	The x-axis scale factor. You can use this property to stretch or shrink an object horizontally.

(usage)	<CompositeTransform />
ScaleY	x:Double
(description)	The y-axis scale factor. You can use this property to stretch or shrink an object vertically.
SkewX	x:Double
(description)	The x-axis skew angle, which is measured in degrees counterclockwise from the y-axis. A skew transform can be useful for creating the illusion of three-dimensional depth in a two-dimensional object.
SkewY	x:Double
(description)	The y-axis skew angle, which is measured in degrees counterclockwise from the x-axis. A skew transform can be useful for creating the illusion of three-dimensional depth in a two-dimensional object.
TranslateX	x:Double
(description)	The distance to translate along the x-axis.
TranslateY	x:Double
(description)	The distance to translate (move) an object along the y-axis.

ContentControl

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl

ButtonBase ListBoxItem ScrollViewer ToolTip	
(usage)	<ContentControl>x:Object</ContentControl>
(description)	Represents a control with a single piece of content. Controls such as Button, CheckBox, and ScrollViewer directly or indirectly inherit from this type.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
Content	x:Object
(description)	The value of the ContentControl property.
ContentTemplate	DataTemplate
(description)	The data template that is used to display the content of the ContentControl.

ContentKeyType (4)

x:Object > ContentKeyType

(usage)	**Aes128Bit**| **Cocktail**
(description)	Represents the content key type.
[is nullable]	False
[text syntax]	ContentKeyTypeSyntax

ContentPresenter

x:Object > DependencyObject > UIElement > FrameworkElement > ContentPresenter

ScrollContentPresenter	
(usage)	<ContentPresenter>x:Object</ContentPresenter>
(description)	Displays the content of a ContentPresenter.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
Content	x:Object
(description)	The data that is used to generate the child elements of a ContentPresenter.
ContentTemplate	DataTemplate
(description)	The template that is used to display the content of the control.

Control

x:Object > DependencyObject > UIElement > FrameworkElement > Control

ContentControl ItemsControl PasswordBox RangeBase RichTextBox TextBox Thumb UserControl	
(usage)	None.
(description)	Represents the base type for UI elements that use a ControlTemplate to define their appearance.
[is default constructible]	False
[name property]	Name
[xml lang property]	Language
(properties)	
Background	Brush
(description)	A brush that provides the background of the control.

ContentControl ItemsControl PasswordBox RangeBase RichTextBox TextBox Thumb UserControl	
BorderBrush	Brush
(description)	A brush that describes the border background of a control.
BorderThickness	Thickness
(description)	The border thickness of a control.
FontFamily	FontFamily
(description)	The font used to display text in the control.
FontSize	x:Double
(description)	The size of the text in this control.
FontStretch	FontStretch
(description)	The degree to which a font is condensed or expanded on the screen.
FontStyle	FontStyle
(description)	The style in which the text is rendered.
FontWeight	FontWeight
(description)	The thickness of the specified font.
Foreground	Brush
(description)	A brush that describes the foreground color.
HorizontalContentAlignment	HorizontalAlignment
(description)	The horizontal alignment of the control's content.
IsEnabled	x:Boolean
(description)	A value indicating whether the user can interact with the control.
IsTabStop	x:Boolean
(description)	A value that indicates whether a control is included in tab navigation.
Padding	Thickness
(description)	The padding inside a control.
TabIndex	x:Int32
(description)	A value that determines the order in which elements receive focus when the user navigates through controls by using the TAB key.
TabNavigation	KeyboardNavigationMode
(description)	A value that modifies how tabbing and TabIndex work for this control.
Template	ControlTemplate

ContentControl ItemsControl PasswordBox RangeBase RichTextBox TextBox Thumb UserControl	
(description)	A control template.
VerticalContentAlignment	VerticalAlignment
(description)	The vertical alignment of the control's content.
(events)	
IsEnabledChanged	Occurs when the IsEnabled property changes.

ControlTemplate

x:Object > DependencyObject > FrameworkTemplate > ControlTemplate

(usage)	<ControlTemplate>**FrameworkElement**</ControlTemplate>
(description)	Defines the element tree that is applied as a control template.
(used by)	Control
[content property]	Template
[dictionary key property]	TargetType
(properties)	
TargetType	x:XamlType
(description)	The type to which the ControlTemplate is applied.

CornerRadius

x:Object > CornerRadius

(usage)	<CornerRadius> string </CornerRadius>
(description)	Describes the characteristics of a rounded corner, such as can be applied to a Border.
(used by)	Border
[is nullable]	False
[text syntax]	CornerRadiusSyntax
(properties)	
BottomLeft	x:Double
(description)	The radius of rounding, in pixels, of the bottom left corner of the object where a CornerRadius is applied.
BottomRight	x:Double
(description)	The radius of rounding, in pixels, of the bottom right corner of the object where a

(usage)	<CornerRadius> string </CornerRadius>
	CornerRadius is applied.
TopLeft	x:Double
(description)	The radius of rounding, in pixels, of the top left corner of the object where a CornerRadius is applied.
TopRight	x:Double
(description)	The radius of rounding, in pixels, of the top right corner of the object where a CornerRadius is applied.

CrossDomainAccess

x:Object > CrossDomainAccess

(usage)	**NoAccess**| **ScriptableOnly**
(description)	Defines values that specify the access that cross-domain callers have to a Silverlight-based application.
[is nullable]	False
[text syntax]	CrossDomainAccessSyntax

CubicEase

x:Object > DependencyObject > EasingFunctionBase > CubicEase, IEasingFunction

(usage)	<CubicEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using the formula f(t) = t3.

Cursor

x:Object > Cursor

(usage)	**Arrow**| **Eraser**| **Hand**| **IBeam**| **None**| **SizeNESW**| **SizeNS**| **SizeNWSE**| **SizeWE**...
(description)	Represents the image used for the mouse pointer.
(used by)	FrameworkElement
[is default constructible]	False
[text syntax]	CursorSyntax

Cursors

x:Object > Cursors

(usage)	<Cursors> string </Cursors>
(description)	Defines a set of default mouse pointer images.
[is default constructible]	False
[text syntax]	CursorsSyntax
(static properties)	
Arrow	Cursor
(description)	Represents an Arrow Cursor.
Eraser	Cursor
(description)	Represents an Eraser Cursor.
Hand	Cursor
(description)	Represents a Hand Cursor.
IBeam	Cursor
(description)	Represents an IBeam Cursor, which is typically used to show where the text cursor appears when the mouse is clicked.
None	Cursor
(description)	Represents a special Cursor that is invisible.
SizeNESW (4)	Cursor
(description)	Represents a SizeNESW Cursor.
SizeNS	Cursor
(description)	Represents a SizeNS Cursor.
SizeNWSE (4)	Cursor
(description)	Represents a SizeNWSE Cursor.
SizeWE	Cursor
(description)	Represents a SizeWE Cursor.
Stylus	Cursor
(description)	Represents a Stylus Cursor.
Wait	Cursor
(description)	Represents a WaitCursor.

DataObject (4)

x:Object > DataObject

(usage)	<DataObject />
(description)	Provides a basic implementation of the IDataObject interface, which defines a format-independent mechanism for transferring data.

DataTemplate

x:Object > DependencyObject > FrameworkTemplate > DataTemplate

(usage)	<DataTemplate>**FrameworkElement**</DataTemplate>
(description)	Describes the visual structure of a data object.
(used by)	ComboBox ContentControl ContentPresenter ItemsControl
[content property]	Template

DeepZoomImageTileSource

x:Object > DependencyObject > MultiScaleTileSource > DeepZoomImageTileSource

(usage)	<DeepZoomImageTileSource> string </DeepZoomImageTileSource>
(description)	Used to specify the source of a MultiScaleImage.
[text syntax]	x:Uri, from [MS-XAML]
(properties)	
UriSource	x:Uri
(description)	The source Uri of the DeepZoomImageTileSource.

DependencyObject

x:Object > DependencyObject

AssemblyPart Brush CacheMode CaptureSource CollectionViewSource ColorKeyFrame ColumnDefinition DependencyObjectCollection(DependencyObjectCollection+T) Deployment DoubleKeyFrame DrawingAttributes EasingFunctionBase Effect ExternalPart FrameworkTemplate GeneralTransform Geometry GradientStop Icon ImageSource InputMethod InputScope InputScopeName KeySpline MultiScaleSubImage MultiScaleTileSource NotificationWindow ObjectKeyFrame OutOfBrowserSettings PathFigure PathSegment PixelShader PointKeyFrame PresentationFrameworkCollection(PresentationFrameworkCollection+T) Projection ResourceDictionary RowDefinition SecuritySettings SetterBase Stroke Style TextElement Timeline TimelineMarker TouchDevice TouchPoint TriggerAction TriggerBase UIElement VisualState VisualStateGroup VisualStateManager VisualTransition Window WindowSettings	
(usage)	None.
(description)	An important foundational Type.
(used by)	Storyboard
[is default constructible]	False

DependencyObjectCollection(T) (4)

x:Object > DependencyObject > DependencyObjectCollection(T)

(usage)	<DependencyObjectCollection x:TypeArguments="T">T*</DependencyObjectCollection>
(description)	Represents a collection of DependencyObject instances of a specified type.
[is list]	True
[allowed types]	T
(events)	
CollectionChanged	Occurs when items in the collection are added, removed, or replaced.

DependencyPropertyChangedEventArgs

x:Object > DependencyPropertyChangedEventArgs

(usage)	<DependencyPropertyChangedEventArgs />
(description)	Provides data for a PropertyChangedCallback implementation.
[is nullable]	False

Deployment

x:Object > DependencyObject > Deployment

(usage)	<Deployment />
(description)	Provides application part and localization information in the application manifest when deploying a Silverlight-based application.
(properties)	
ExternalParts	ExternalPartCollection
(description)	A collection of ExternalPart instances that represent the external assemblies required by the application.
[read only]	True
Parts	AssemblyPartCollection
(description)	A collection of assembly parts that are included in the deployment.
[read only]	True
(static properties)	
Current	Deployment
(description)	The current Deployment object.

DiscreteColorKeyFrame

x:Object > DependencyObject > ColorKeyFrame > DiscreteColorKeyFrame

(usage)	<DiscreteColorKeyFrame />
(description)	Animates from the Color value of the previous key frame to its own Value using discrete values.

DiscreteDoubleKeyFrame

x:Object > DependencyObject > DoubleKeyFrame > DiscreteDoubleKeyFrame

(usage)	<DiscreteDoubleKeyFrame />
(description)	Animates from the Double value of the previous key frame to its own Value using discrete values.

DiscreteObjectKeyFrame

x:Object > DependencyObject > ObjectKeyFrame > DiscreteObjectKeyFrame

(usage)	<DiscreteObjectKeyFrame />
(description)	Animates from the Object value of the previous key frame to its own Value using discrete values.

DiscretePointKeyFrame

x:Object > DependencyObject > PointKeyFrame > DiscretePointKeyFrame

(usage)	<DiscretePointKeyFrame />
(description)	Animates from the Point value of the previous key frame to its own Value using discrete frames.

DockPosition

x:Object > DockPosition

(usage)	**Top**| **Left**| **Bottom**| **Right**| **Fill**| **None**
(description)	Contains values that specify the dock position of an object within a docking container.
[is nullable]	False
[text syntax]	DockPositionSyntax

DomainAcquirer (4)

x:Object > DomainAcquirer

(usage)	<DomainAcquirer />

(usage)	<DomainAcquirer />
(description)	Handles Join Domain requests.
(used by)	LicenseAcquirer
(properties)	
ChallengeCustomData	x:String
(description)	A string containing service specific data to be conveyed to the domain server without implementing manual domain join/leave.
(events)	
JoinDomainCompleted	Occurs when the join domain operation completes.
LeaveDomainCompleted	Occurs when the leave domain operation completes.

DoubleAnimation

x:Object > DependencyObject > Timeline > DoubleAnimation

(usage)	<DoubleAnimation />
(description)	Animates the value of a Double property between two target values using linear interpolation over a specified Duration.
(properties)	
By	x:Nullable(x:Double)
(description)	The total amount by which the animation changes its starting value.
EasingFunction	IEasingFunction
(description)	The easing function applied to this animation.
From	x:Nullable(x:Double)
(description)	The animation's starting value.
To	x:Nullable(x:Double)
(description)	The animation's ending value.

DoubleAnimationUsingKeyFrames

x:Object > DependencyObject > Timeline > DoubleAnimationUsingKeyFrames

(usage)	<DoubleAnimationUsingKeyFrames>**DoubleKeyFrame***</DoubleAnimationUsingKeyFrames>
(description)	Animates the value of a Double property along a set of KeyFrames.
[content property]	KeyFrames

(usage)	<DoubleAnimationUsingKeyFrames>**DoubleKeyFrame***</DoubleAnimationUsingKeyFrames>
(properties)	
KeyFrames	DoubleKeyFrameCollection
(description)	The collection of DoubleKeyFrame objects that define the animation.
[read only]	True

DoubleCollection

x:Object > DependencyObject > PresentationFrameworkCollection(x:Double) > DoubleCollection

(usage)	<DoubleCollection> string </DoubleCollection>
(description)	Represents an ordered collection of Double values.
(used by)	Shape
[text syntax]	DoubleCollectionSyntax
[is list]	True
[allowed types]	x:Double

DoubleKeyFrame

x:Object > DependencyObject > DoubleKeyFrame

DiscreteDoubleKeyFrame EasingDoubleKeyFrame LinearDoubleKeyFrame SplineDoubleKeyFrame	
(usage)	None.
(description)	A base type that defines an animation segment with its own target value and interpolation technique for a DoubleAnimationUsingKeyFrames.
(used by)	DoubleKeyFrameCollection
[is default constructible]	False
(properties)	
KeyTime	KeyTime
(description)	The time at which the key frame's target Value should be reached.
Value	x:Double
(description)	The key frame's target value.

DoubleKeyFrameCollection

x:Object > DependencyObject > PresentationFrameworkCollection(DoubleKeyFrame) > DoubleKeyFrameCollection

(usage)	<DoubleKeyFrameCollection>**DoubleKeyFrame***</DoubleKeyFrameCollection>
(description)	Represents a collection of DoubleKeyFrame objects that can be individually accessed by index.
(used by)	DoubleAnimationUsingKeyFrames
[is list]	True
[allowed types]	DoubleKeyFrame

DrawingAttributes

x:Object > DependencyObject > DrawingAttributes

(usage)	<DrawingAttributes />
(description)	Specifies drawing attributes that are used to draw a Stroke.
(used by)	Stroke
(properties)	
Color	Color
(description)	The color that is used to draw a Stroke.
Height	x:Double
(description)	The height of the stylus that is used to draw a Stroke.
OutlineColor	Color
(description)	The outline color that is used to draw a Stroke.
Width	x:Double
(description)	The width of the stylus that is used to draw a Stroke.

DropShadowEffect

x:Object > DependencyObject > Effect > DropShadowEffect

(usage)	<DropShadowEffect />
(description)	Applies a shadow behind a visual object at a slight offset. The offset is determined by mimicking a casting shadow from an imaginary light source.
(properties)	
BlurRadius	x:Double
(description)	How defined the edges of the shadow are (how blurry the shadow is).
Color	Color

(usage)	<DropShadowEffect />
(description)	The color of the shadow.
Direction	x:Double
(description)	The angle at which the shadow is cast.
Opacity	x:Double
(description)	The degree of opacity of the shadow.
ShadowDepth	x:Double
(description)	The distance between the object and the shadow that it casts.

Duration

x:Object > Duration

(usage)	<Duration> string </Duration>
(description)	Represents the duration of time that a Timeline is active.
(used by)	Timeline VisualTransition
[is nullable]	False
[text syntax]	DurationSyntax
(static properties)	
Automatic	Duration
(description)	A Duration value that is automatically determined.
Forever	Duration
(description)	A Duration value that represents an infinite interval.

EasingColorKeyFrame

x:Object > DependencyObject > ColorKeyFrame > EasingColorKeyFrame

(usage)	<EasingColorKeyFrame />
(description)	A type that enables you to associate easing functions with a ColorAnimationUsingKeyFrames key frame animation.
(properties)	
EasingFunction	IEasingFunction
(description)	The easing function that is applied to the key frame.

EasingDoubleKeyFrame

x:Object > DependencyObject > DoubleKeyFrame > EasingDoubleKeyFrame

(usage)	<EasingDoubleKeyFrame />
(description)	Defines a property that enables you to associate an easing function with a DoubleAnimationUsingKeyFrames key-frame animation.
(properties)	
EasingFunction	IEasingFunction
(description)	The easing function that is applied to the key frame.

EasingFunctionBase

x:Object > DependencyObject > EasingFunctionBase, IEasingFunction

BackEase BounceEase CircleEase CubicEase ElasticEase ExponentialEase PowerEase QuadraticEase QuarticEase QuinticEase SineEase	
(usage)	None.
(description)	Provides the base type for all the easing functions. You can create your own custom easing functions by inheriting from this type.
[is default constructible]	False
(properties)	
EasingMode	EasingMode
(description)	A value that specifies how the animation interpolates.

EasingMode

x:Object > EasingMode

(usage)	**EaseOut**| **EaseIn**| **EaseInOut**
(description)	Specifies how the animation associated with an easing function interpolates.
(used by)	EasingFunctionBase
[is nullable]	False
[text syntax]	EasingModeSyntax

EasingPointKeyFrame

x:Object > DependencyObject > PointKeyFrame > EasingPointKeyFrame

(usage)	<EasingPointKeyFrame />

(usage)	<EasingPointKeyFrame />
(description)	Defines a property that enables you to associate an easing function with a PointAnimationUsingKeyFrames key-frame animation.
(properties)	
EasingFunction	IEasingFunction
(description)	The easing function that is applied to the key frame.

Effect

x:Object > DependencyObject > Effect

BlurEffect DropShadowEffect	
(usage)	{x:Static Effect.StaticPropertyName}
(description)	Provides a base type for all bitmap effects.
(used by)	UIElement
[is default constructible]	False
(static properties)	
ImplicitInput	Brush
(description)	A Brush that, when it is used as an input for an Effect, causes the bitmap of the UIElement that the Effect is applied to be that input.

ElasticEase

x:Object > DependencyObject > EasingFunctionBase > ElasticEase, IEasingFunction

(usage)	<ElasticEase />
(description)	Represents an easing function that creates an animation that resembles a spring oscillating back and forth until it comes to rest.
(properties)	
Oscillations	x:Int32
(description)	The number of times the target slides back and forth over the animation destination.
Springiness	x:Double
(description)	The stiffness of the spring. The smaller the Springiness value is, the stiffer the spring and the faster the elasticity decreases in intensity over each oscillation.

ElevatedPermissions (4)

x:Object > ElevatedPermissions

(usage)	**NotRequired**| **Required**
(description)	Defines constants that indicate whether elevated permissions are required for an out-of-browser application.
[is nullable]	False
[text syntax]	ElevatedPermissionsSyntax

Ellipse

x:Object > DependencyObject > UIElement > FrameworkElement > Shape > Ellipse

(usage)	<Ellipse />
(description)	Draws an ellipse.
[name property]	Name
[xml lang property]	Language

EllipseGeometry

x:Object > DependencyObject > Geometry > EllipseGeometry

(usage)	<EllipseGeometry />
(description)	Represents the geometry of a circle or ellipse.
(properties)	
Center	Point
(description)	The center point of the EllipseGeometry.
RadiusX	x:Double
(description)	The x-radius value of the EllipseGeometry.
RadiusY	x:Double
(description)	The y-radius value of the EllipseGeometry.

EventTrigger

x:Object > DependencyObject > TriggerBase > EventTrigger

(usage)	<EventTrigger>**TriggerAction***</EventTrigger>
(description)	Represents a trigger that applies a set of actions (animation storyboards) in response to an event.
[content property]	Actions

(usage)	<EventTrigger>**TriggerAction***</EventTrigger>
(properties)	
Actions	TriggerActionCollection
(description)	The collection of BeginStoryboard objects that this EventTrigger maintains.
[read only]	True

ExpandCollapseState

x:Object > ExpandCollapseState

(usage)	**Collapsed**| **Expanded**| **PartiallyExpanded**| **LeafNode**
(description)	Contains values that specify the ExpandCollapseState automation property value of a UI automation element.
[is nullable]	False
[text syntax]	ExpandCollapseStateSyntax

ExponentialEase

x:Object > DependencyObject > EasingFunctionBase > ExponentialEase, IEasingFunction

(usage)	<ExponentialEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using an exponential formula (see remarks).
(properties)	
Exponent	x:Double
(description)	The exponent used to determine the interpolation of the animation.

ExtensionPart

x:Object > DependencyObject > ExternalPart > ExtensionPart

(usage)	<ExtensionPart />
(description)	Represents a zip file containing assemblies used by application library caching.
(properties)	
Source	x:Uri
(description)	The file name or URI of the external library package.

ExternalPart

x:Object > DependencyObject > ExternalPart

ExtensionPart	
(usage)	<ExternalPart />
(description)	Defines a base type for specifying parts of a Silverlight application that are external to the application package (.xap file).
(used by)	ExternalPartCollection
(properties)	
Source (4)	x:Uri
(description)	The URI of the external part.

ExternalPartCollection

x:Object > DependencyObject > PresentationFrameworkCollection(ExternalPart) > ExternalPartCollection

(usage)	<ExternalPartCollection>**ExternalPart***</ExternalPartCollection>
(description)	Represents a collection of ExternalPart instances that indicate parts of a Silverlight application that are external to the application package (.xap file).
(used by)	Deployment
[is list]	True
[allowed types]	ExternalPart

FillBehavior

x:Object > FillBehavior

(usage)	**HoldEnd**| **Stop**
(description)	Specifies how a Timeline behaves when it is outside its active period but its parent is inside its active or hold period.
(used by)	Timeline
[is nullable]	False
[text syntax]	FillBehaviorSyntax

FillRule

x:Object > FillRule

(usage)	**EvenOdd**| **Nonzero**
(description)	Specifies how the intersecting areas of PathFigure objects contained in a Geometry are combined to form the area of the Geometry.
(used by)	GeometryGroup PathGeometry Polygon Polyline
[is nullable]	False
[text syntax]	FillRuleSyntax

FlowDirection (4)

x:Object > FlowDirection

(usage)	**LeftToRight**| **RightToLeft**
(description)	Defines constants that specify the content flow direction for text and UI elements.
(used by)	FrameworkElement Run
[is nullable]	False
[text syntax]	FlowDirectionSyntax

FontFamily

x:Object > FontFamily

(usage)	<FontFamily> string </FontFamily>
(description)	Represents a family of related fonts.
(used by)	Control Inline TextBlock TextElement
[is default constructible]	False
[text syntax]	FontFamilySyntax

Fonts (4)

x:Object > Fonts

(usage)	None.
(description)	Provides enumeration support for FontFamily and Typeface objects.
[is default constructible]	False

FontStretch

x:Object > FontStretch

(usage)	**Condensed**| **Expanded**| **ExtraCondensed**| **ExtraExpanded**| **Normal**| **SemiCondensed**| **SemiExpanded**| **UltraCondensed**| **UltraExpanded**...
(description)	Describes the degree to which a font has been stretched, compared to the normal aspect ratio of that font.
(used by)	Control Inline TextBlock TextElement
[is nullable]	False
[text syntax]	FontStretchSyntax

FontStretches

x:Object > FontStretches

(usage)	{x:Static FontStretches.StaticPropertyName}
(description)	Provides a set of predefined font stretches as static property values.
[is default constructible]	False
(static properties)	
Condensed	FontStretch
(description)	Specifies a condensed font stretch.
Expanded	FontStretch
(description)	Specifies an expanded font stretch.
ExtraCondensed	FontStretch
(description)	Specifies an extra-condensed font stretch.
ExtraExpanded	FontStretch
(description)	Specifies an extra-expanded font stretch.
Normal	FontStretch
(description)	Specifies a normal font stretch.
SemiCondensed	FontStretch
(description)	Specifies a semi-condensed font stretch.
SemiExpanded	FontStretch
(description)	Specifies a semi-expanded font stretch.
UltraCondensed	FontStretch
(description)	Specifies an ultra-condensed font stretch.
UltraExpanded	FontStretch

(usage)	{x:Static FontStretches.StaticPropertyName}
(description)	Specifies an ultra-expanded font stretch.

FontStyle

x:Object > FontStyle

(usage)	**Italic**| **Normal**
(description)	Represents the style of a font face (for instance, as normal or italic).
(used by)	Control Inline TextBlock TextElement
[is nullable]	False
[text syntax]	FontStyleSyntax

FontStyles

x:Object > FontStyles

(usage)	{x:Static FontStyles.StaticPropertyName}
(description)	Provides a set of predefined font styles as static property values.
[is default constructible]	False
(static properties)	
Italic	FontStyle
(description)	Specifies an italic font style.
Normal	FontStyle
(description)	Specifies a normal, or roman, font style.

FontWeight

x:Object > FontWeight

(usage)	**Black**| **Bold**| **ExtraBlack**| **ExtraBold**| **ExtraLight**| **Light**| **Medium**| **Normal**| **SemiBold**...
(description)	Refers to the density of a typeface, in terms of the lightness or heaviness of the strokes.
(used by)	Control Inline TextBlock TextElement
[is nullable]	False
[text syntax]	FontWeightSyntax

FontWeights

x:Object > FontWeights

(usage)	{x:Static FontWeights.StaticPropertyName}
(description)	Provides a set of predefined font weights as static property values.
[is default constructible]	False
(static properties)	
Black	FontWeight
(description)	Specifies a "Black" font weight.
Bold	FontWeight
(description)	Specifies a "Bold" font weight.
ExtraBlack	FontWeight
(description)	Specifies an "ExtraBlack" font weight.
ExtraBold	FontWeight
(description)	Specifies an "ExtraBold" font weight.
ExtraLight	FontWeight
(description)	Specifies an "ExtraLight" font weight.
Light	FontWeight
(description)	Specifies a "Light" font weight.
Medium	FontWeight
(description)	Specifies a "Medium" font weight.
Normal	FontWeight
(description)	Specifies a "Normal" font weight.
SemiBold	FontWeight
(description)	Specifies a "SemiBold" font weight.
Thin	FontWeight
(description)	Specifies a "Thin" font weight.

FrameworkElement

x:Object > DependencyObject > UIElement > FrameworkElement

Border **ContentPresenter** **Control** **Glyphs** **Image** **ItemsPresenter** **MediaElement** **MultiScaleImage** **Panel** **Popup** **Shape** **TextBlock** **Viewbox** **WebBrowser**

Border ContentPresenter Control Glyphs Image ItemsPresenter MediaElement MultiScaleImage Panel Popup Shape TextBlock Viewbox WebBrowser	
(usage)	None.
(description)	An object that participates in layout, data binding, and object tree.
(used by)	NotificationWindow
[is default constructible]	False
[name property]	Name
[xml lang property]	Language
(properties)	
Cursor	Cursor
(description)	The cursor that displays while the mouse pointer is over a FrameworkElement.
DataContext	x:Object
(description)	The data context for a FrameworkElement when it participates in data binding.
FlowDirection (4)	FlowDirection
(description)	The direction that text and other user interface elements flow within any parent element that controls their layout.
Height	x:Double
(description)	The suggested height of a FrameworkElement.
[text syntax]	LengthSyntax
HorizontalAlignment	HorizontalAlignment
(description)	The horizontal alignment characteristics that are applied to a FrameworkElement when it is composed in a layout parent, such as a panel or items control.
Margin	Thickness
(description)	The outer margin of a FrameworkElement.
MaxHeight	x:Double
(description)	The maximum height constraint of a FrameworkElement.
MaxWidth	x:Double
(description)	The maximum width constraint of a FrameworkElement.
MinHeight	x:Double
(description)	The minimum height constraint of a FrameworkElement.

Border ContentPresenter Control Glyphs Image ItemsPresenter MediaElement MultiScaleImage Panel Popup Shape TextBlock Viewbox WebBrowser	
MinWidth	x:Double
(description)	The minimum width constraint of a FrameworkElement.
Name	x:String
(description)	The identifying name of the object.
Resources	ResourceDictionary
(description)	The locally defined resource dictionary. In XAML, you can establish resource items as child object elements of a frameworkElement.Resources property element, through XAML implicit collection syntax.
Style	Style
(description)	An instance Style that is applied for this object during rendering.
Tag	x:Object
(description)	An arbitrary object value that can be used to store custom information about this object.
Triggers	TriggerCollection
(description)	The collection of triggers for animations that are defined for a FrameworkElement.
[read only]	True
VerticalAlignment	VerticalAlignment
(description)	The vertical alignment characteristics that are applied to a FrameworkElement when it is composed in a parent object such as a panel or items control.
Width	x:Double
(description)	The width of a FrameworkElement.
[text syntax]	LengthSyntax
(events)	
BindingValidationError	Occurs when a data validation error is reported by a binding source.
LayoutUpdated	Occurs when the layout of the Silverlight visual tree changes.
Loaded	Occurs when a FrameworkElement has been constructed and added to the object tree.
SizeChanged	Occurs when either the ActualHeight or the ActualWidth properties change value on a FrameworkElement.
Unloaded (4)	Occurs when this object is no longer connected to the main object tree.

FrameworkTemplate

x:Object > DependencyObject > FrameworkTemplate

ControlTemplate DataTemplate ItemsPanelTemplate	
(usage)	None.
(description)	Creates an element tree of elements.
[is default constructible]	False
[content property]	Template
(properties)	
Template	FrameworkElement
(description)	Defines the root element for the Template.

GeneralTransform

x:Object > DependencyObject > GeneralTransform

Transform	
(usage)	None.
(description)	Provides generalized transformation support for objects, such as points and rectangles.
[is default constructible]	False

GeneratorDirection

x:Object > GeneratorDirection

(usage)	**Forward**| **Backward**
(description)	Specifies the direction in which item generation will occur.
[is nullable]	False
[text syntax]	GeneratorDirectionSyntax

GeneratorPosition

x:Object > GeneratorPosition

(usage)	<GeneratorPosition />
(description)	GeneratorPosition is used to describe the position of an item that is managed by ItemContainerGenerator.
[is nullable]	False

(usage)	<GeneratorPosition />
(properties)	
Index	x:Int32
(description)	The Int32 index that is relative to the generated (realized) items.
Offset	x:Int32
(description)	The Int32 offset that is relative to the ungenerated (unrealized) items near the indexed item.

Geometry

x:Object > DependencyObject > Geometry

EllipseGeometry GeometryGroup LineGeometry PathGeometry RectangleGeometry	
(usage)	<Geometry> string </Geometry>
(description)	Provides a base type for objects that define geometric shapes. Geometry objects can be used for clipping regions and as geometry definitions for rendering two-dimensional graphic data as a Path.
(used by)	GeometryCollection Path UIElement
[is default constructible]	False
[text syntax]	GeometrySyntax
(properties)	
Transform	Transform
(description)	The Transform object applied to a Geometry.
(static properties)	
Empty	Geometry
(description)	An empty geometry object.
StandardFlatteningTolerance	x:Double
(description)	The standard tolerance used for polygonal approximation.

GeometryCollection

x:Object > DependencyObject > PresentationFrameworkCollection(Geometry) > GeometryCollection

(usage)	<GeometryCollection>**Geometry***</GeometryCollection>
(description)	Represents a collection of Geometry objects.
(used by)	GeometryGroup

(usage)	<GeometryCollection>**Geometry***</GeometryCollection>
[is list]	True
[allowed types]	Geometry

GeometryGroup

x:Object > DependencyObject > Geometry > GeometryGroup

(usage)	<GeometryGroup>**Geometry***</GeometryGroup>
(description)	Represents a composite geometry, composed of other Geometry objects.
[content property]	Children
(properties)	
Children	GeometryCollection
(description)	The GeometryCollection that contains the objects that define this GeometryGroup.
FillRule	FillRule
(description)	How the intersecting areas of the objects contained in this GeometryGroup are combined.

Glyphs

x:Object > DependencyObject > UIElement > FrameworkElement > Glyphs

(usage)	<Glyphs />
(description)	Provides a visual representation of letters, characters, or symbols, in a specific font and style.
[name property]	Name
[xml lang property]	Language
(properties)	
Fill	Brush
(description)	The Brush that is used to render the glyphs.
FontRenderingEmSize	x:Double
(description)	The em size used for rendering the glyphs.
FontUri	x:Uri
(description)	The location of the font used for rendering the glyphs.
Indices	x:String

(usage)	<Glyphs />
(description)	The glyph indices for the glyphs.
OriginX	x:Double
(description)	The x origin for the glyphs.
OriginY	x:Double
(description)	The y origin for the glyphs.
StyleSimulations	StyleSimulations
(description)	The style simulations applied to the glyphs.
UnicodeString	x:String
(description)	The Unicode string to render in glyphs.

GradientBrush

x:Object > DependencyObject > Brush > GradientBrush

LinearGradientBrush RadialGradientBrush	
(usage)	None.
(description)	A base type that describes a gradient, composed of gradient stops. Types that derive from GradientBrush describe different ways of interpreting gradient stops.
[is default constructible]	False
[content property]	GradientStops
(properties)	
ColorInterpolationMode	ColorInterpolationMode
(description)	A ColorInterpolationMode enumeration value that specifies how the gradient's colors are interpolated.
GradientStops	GradientStopCollection
(description)	The brush's gradient stops.
MappingMode	BrushMappingMode
(description)	A BrushMappingMode enumeration value that specifies whether the positioning coordinates of the gradient brush are absolute or relative to the output area.
SpreadMethod	GradientSpreadMethod
(description)	The type of spread method that specifies how to draw a gradient that starts or ends inside the bounds of the object to be painted.

GradientSpreadMethod

x:Object > GradientSpreadMethod

(usage)	**Pad**| **Reflect**| **Repeat**
(description)	Specifies how to draw the gradient outside a gradient brush's gradient vector or space.
(used by)	GradientBrush
[is nullable]	False
[text syntax]	GradientSpreadMethodSyntax

GradientStop

x:Object > DependencyObject > GradientStop

(usage)	<GradientStop>**Color**</GradientStop>
(description)	Describes the location and color of a transition point in a gradient.
(used by)	GradientStopCollection
[content property]	Color
(properties)	
Color	Color
(description)	The color of the gradient stop.
Offset	x:Double
(description)	The location of the gradient stop within the gradient vector.

GradientStopCollection

x:Object > DependencyObject > PresentationFrameworkCollection(GradientStop) > GradientStopCollection

(usage)	<GradientStopCollection>**GradientStop***</GradientStopCollection>
(description)	Represents a collection of GradientStop objects that can be individually accessed by index.
(used by)	GradientBrush
[is list]	True
[allowed types]	GradientStop

Grid

x:Object > DependencyObject > UIElement > FrameworkElement > Panel > Grid

(usage)	<Grid>**UIElement***</Grid>
(description)	Defines a flexible grid area that consists of columns and rows.
[content property]	Children
[name property]	Name
[xml lang property]	Language
(properties)	
ColumnDefinitions	ColumnDefinitionCollection
(description)	A ColumnDefinitionCollection defined on this instance of Grid.
[read only]	True
RowDefinitions	RowDefinitionCollection
(description)	A RowDefinitionCollection defined on this instance of Grid.
[read only]	True
ShowGridLines	x:Boolean
(description)	A value that indicates whether grid lines are visible within this Grid.
(attachable properties)	
Grid.Column	x:Int32
(description)	A value that indicates which column child content within a Grid should appear in.
[target type]	FrameworkElement
Grid.ColumnSpan	x:Int32
(description)	A value that indicates the total number of columns that child content spans within a Grid.
[target type]	FrameworkElement
Grid.Row	x:Int32
(description)	A value that indicates which row child content within a Grid should appear in.
[target type]	FrameworkElement
Grid.RowSpan	x:Int32
(description)	A value that indicates the total number of rows that child content spans within a Grid.
[target type]	FrameworkElement

GridLength

x:Object > GridLength

(usage)	<GridLength> string </GridLength>
(description)	Represents the length of elements that explicitly support Star unit types.
(used by)	ColumnDefinition RowDefinition
[is nullable]	False
[text syntax]	GridLengthSyntax
(static properties)	
Auto	GridLength
(description)	An instance of GridLength that holds a value whose size is determined by the size properties of the content object.

GridUnitType

x:Object > GridUnitType

(usage)	**Auto**| **Pixel**| **Star**
(description)	Describes the kind of value that a GridLength object is holding.
[is nullable]	False
[text syntax]	GridUnitTypeSyntax

HorizontalAlignment

x:Object > HorizontalAlignment

(usage)	**Left**| **Center**| **Right**| **Stretch**
(description)	Indicates where an element should be displayed on the horizontal axis relative to the allocated layout slot of the parent element.
(used by)	Control FrameworkElement
[is nullable]	False
[text syntax]	HorizontalAlignmentSyntax

Hyperlink (4)

x:Object > DependencyObject > TextElement > Inline > Span > Hyperlink

(usage)	<Hyperlink>[**Inline**|**x:String**|**UIElement**]*</Hyperlink>

(usage)	<Hyperlink>[**Inline**|**x:String**|**UIElement**]*</Hyperlink>
(description)	Provides an inline-level content element that provides facilities for hosting hyperlinks.
[content property]	Inlines
[xml lang property]	Language
(properties)	
CommandParameter	x:Object
(description)	Command parameters associated with the command specified by the Command property.
MouseOverForeground	Brush
(description)	The brush that paints the foreground color when the mouse pointer moves over the Hyperlink.
MouseOverTextDecorations	TextDecorationCollection
(description)	The TextDecorationCollection that decorates the Hyperlink.
NavigateUri	x:Uri
(description)	A URI to navigate to when the Hyperlink is activated.
TargetName	x:String
(description)	The name of a target window or frame for the Hyperlink.
(events)	
Click	Occurs when the left mouse button is clicked on a Hyperlink.

HyperlinkButton

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase > HyperlinkButton

(usage)	<HyperlinkButton>**x:Object**</HyperlinkButton>
(description)	Represents a button control that displays a hyperlink.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
NavigateUri	x:Uri

(usage)	<HyperlinkButton>**x:Object**</HyperlinkButton>
(description)	The URI to navigate to when the HyperlinkButton is clicked.
[text syntax]	x:Uri, from [MS-XAML]
TargetName	x:String
(description)	The name of the target window or frame that the Web page should open in, or the name of the object within the Silverlight application to navigate to.

Icon

x:Object > DependencyObject > Icon

(usage)	<Icon>**x:Object**</Icon>
(description)	Represents an icon that is used to identify an offline application.
(used by)	IconCollection
[content property]	Source

IconCollection

x:Object > DependencyObject > PresentationFrameworkCollection(Icon) > IconCollection

(usage)	<IconCollection>**Icon***</IconCollection>
(description)	Represents a collection of Icon instances.
(used by)	OutOfBrowserSettings
[is list]	True
[allowed types]	Icon

IDataObject (4)

IDataObject

DataObject	
(usage)	None.
(description)	Provides a format-independent mechanism for transferring data.
[is default constructible]	False

IEasingFunction

IEasingFunction

BackEase BounceEase CircleEase CubicEase EasingFunctionBase ElasticEase ExponentialEase PowerEase QuadraticEase QuarticEase QuinticEase SineEase	
(usage)	None.
(description)	Defines the basic functionality of an easing function.
(used by)	ColorAnimation DoubleAnimation EasingColorKeyFrame EasingDoubleKeyFrame EasingPointKeyFrame PointAnimation VisualTransition
[is default constructible]	False

Image

x:Object > DependencyObject > UIElement > FrameworkElement > Image

(usage)	<Image />
(description)	Represents a control that displays an image in the JPEG or PNG file formats.
[name property]	Name
[xml lang property]	Language
(properties)	
Source	ImageSource
(description)	The source for the image.
Stretch	Stretch
(description)	A value that describes how an Image should be stretched to fill the destination rectangle.
(events)	
ImageFailed	Occurs when there is an error associated with image retrieval or format.
ImageOpened	Occurs when the image source is downloaded and decoded with no failure. You can use this event to determine the size of an image before rendering it.

ImageBrush

x:Object > DependencyObject > Brush > TileBrush > ImageBrush

(usage)	<ImageBrush />
(description)	Paints an area with an image.
(properties)	
ImageSource	ImageSource
(description)	The image displayed by this ImageBrush.

(usage)	<ImageBrush />
(events)	
ImageFailed	Occurs when there is an error associated with image retrieval or format.
ImageOpened	Occurs when the image source is downloaded and decoded with no failure. You can use this event to determine the size of an image before rendering it.

ImageSource

x:Object > DependencyObject > ImageSource

BitmapSource	
(usage)	<ImageSource> string </ImageSource>
(description)	Provides an object source type for Source, Source, and ImageSource.
(used by)	Image ImageBrush
[is default constructible]	False
[text syntax]	x:Uri, from [MS-XAML]

ImeConversionModeValues (4)

x:Object > ImeConversionModeValues

(usage)	**Alphanumeric**| **Native**| **Katakana**| **FullShape**| **Roman**| **CharCode**| **NoConversion**| **Eudc**| **Symbol**...
(description)	Describes a mode of input conversion to be performed by an input method editor interacting with a Silverlight-based application.
(used by)	InputMethod
[is nullable]	False
[text syntax]	ImeConversionModeValuesSyntax

ImplicitInputBrush

x:Object > DependencyObject > Brush > ImplicitInputBrush

(usage)	<ImplicitInputBrush />
(description)	Represents the input bitmap to a shader.

InkPresenter

x:Object > DependencyObject > UIElement > FrameworkElement > Panel > Canvas > InkPresenter

(usage)	<InkPresenter>**UIElement***</InkPresenter>
(description)	Implements a rectangular surface that displays ink strokes.
[content property]	Children
[name property]	Name
[xml lang property]	Language
(properties)	
Strokes	StrokeCollection
(description)	The strokes that the InkPresenter displays.

Inline

x:Object > DependencyObject > TextElement > Inline

InlineUIContainer LineBreak Run Span	
(usage)	None.
(description)	Provides a base for inline flow content element behavior.
(used by)	InlineCollection
[is default constructible]	False
[xml lang property]	Language
(properties)	
FontFamily	FontFamily
(description)	The preferred top-level font family for the content in this element.
FontSize	x:Double
(description)	The font size for the content in this element.
FontStretch	FontStretch
(description)	The glyph width of the font in a family to select.
FontStyle	FontStyle
(description)	The font style for the content in this element.
FontWeight	FontWeight
(description)	The top-level font weight to select from the font family for the content in this element.
Foreground	Brush
(description)	The Brush to apply to the content in this element.

InlineUIContainer LineBreak Run Span	
TextDecorations	TextDecorationCollection
(description)	A value that specifies the text decorations that are applied to the content in an Inline element.

InlineCollection

x:Object > DependencyObject > TextElementCollection(Inline) > InlineCollection

(usage)	None.
(description)	Represents a collection of Inline elements.
(used by)	Paragraph Span TextBlock
[is default constructible]	False
[whitespace significant collection]	True
[is list]	True
[allowed types]	Inline x:String UIElement

InlineUIContainer (4)

x:Object > DependencyObject > TextElement > Inline > InlineUIContainer

(usage)	<InlineUIContainer>**UIElement**</InlineUIContainer>
(description)	Provides an inline content element that enables UIElement types to be embedded in content.
[content property]	Child
[xml lang property]	Language
(properties)	
Child	UIElement
(description)	The UIElement hosted by the InlineUIContainer.

InputMethod

x:Object > DependencyObject > InputMethod

(usage)	None.
(description)	Specifies attached properties that influence input method editor (IME) features and support when used with a Silverlight-based application.

(usage)	None.
[is default constructible]	False
(attachable properties)	
InputMethod.IsInputMethodEnabled	x:Boolean
(description)	A value that determines whether input method editors can be used to provide input for the control where the property is attached.
[target type]	DependencyObject
InputMethod.PreferredImeConversionMode (4)	ImeConversionModeValues
(description)	A preferred ImeConversionModeValues value for a specified dependency object.
[target type]	DependencyObject
InputMethod.PreferredImeState (4)	InputMethodState
(description)	A preferred input method state for a specified dependency object.
[target type]	DependencyObject

InputMethodState (4)

x:Object > InputMethodState

(usage)	**Off**| **On**| **DoNotCare**
(description)	Describes the state of an input method editor when it interacts with a Silverlight-based application.
(used by)	InputMethod
[is nullable]	False
[text syntax]	InputMethodStateSyntax

InputScope (4)

x:Object > DependencyObject > InputScope

(usage)	<InputScope />
(description)	Represents information related to the scope of data provided by an input method.
(used by)	TextBox
(properties)	

(usage)	<InputScope />
Names	IList
(description)	The input scope names, as a list of strings.
[read only]	True

InputScopeName (4)

x:Object > DependencyObject > InputScopeName

(usage)	<InputScopeName>**InputScopeNameValue**</InputScopeName>
(description)	Defines a name for text input patterns.
[content property]	NameValue
(properties)	
NameValue	InputScopeNameValue
(description)	The input scope name value, which modifies how input from alternative input methods is interpreted.

InputScopeNameValue (4)

x:Object > InputScopeNameValue

(usage)	**AddressCity**| **AddressCountryName**| **AddressCountryShortName**| **AddressStateOrProvince**| **AddressStreet**| **AlphanumericFullWidth**| **AlphanumericHalfWidth**| **ApplicationEnd**| **Bopomofo**...
(description)	Specifies the input scope name, which modifies how input from alternative input methods is interpreted.
(used by)	InputScopeName
[is nullable]	False
[text syntax]	InputScopeNameValueSyntax

InstallState

x:Object > InstallState

(usage)	**NotInstalled**| **Installing**| **Installed**| **InstallFailed**
(description)	Defines constants that indicate the installation state of an application that is configured to run outside the browser.
[is nullable]	False
[text	InstallStateSyntax

(usage)	**NotInstalled**| **Installing**| **Installed**| **InstallFailed**
syntax]	

IScrollInfo

IScrollInfo

ScrollContentPresenter **VirtualizingStackPanel**	
(usage)	None.
(description)	Represents the main scrollable region inside a ScrollViewer control.
[is default constructible]	False
(properties)	
CanHorizontallyScroll	x:Boolean
(description)	A value that indicates whether scrolling on the horizontal axis is possible.
CanVerticallyScroll	x:Boolean
(description)	A value that indicates whether scrolling on the vertical axis is possible.
ScrollOwner	ScrollViewer
(description)	A ScrollViewer element that controls scrolling behavior.

Italic (4)

x:Object > DependencyObject > TextElement > Inline > Span > Italic

(usage)	<Italic>[**Inline**|**x:String**|**UIElement**]*</Italic>
(description)	Provides an inline-level flow content element that causes content to appear with an italic font style.
[content property]	Inlines
[xml lang property]	Language

ItemCollection

x:Object > DependencyObject > PresentationFrameworkCollection(x:Object) > ItemCollection

(usage)	None.
(description)	Holds the list of items that represent the content of an ItemsControl.
(used by)	ItemsControl
[is default constructible]	False

(usage)	None.
[is list]	True
[allowed types]	x:Object

ItemsControl

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ItemsControl

Selector	
(usage)	<ItemsControl>x:Object*</ItemsControl>
(description)	Represents a control that can be used to present a collection of items.
[content property]	Items
[name property]	Name
[xml lang property]	Language
(properties)	
DisplayMemberPath	x:String
(description)	The name or path of the property that is displayed for each data item.
Items	ItemCollection
(description)	The collection used to generate the content of the control.
[read only]	True
ItemsPanel	ItemsPanelTemplate
(description)	The template that defines the panel that controls the layout of items.
ItemsSource	IEnumerable
(description)	A collection used to generate the content of the ItemsControl.
ItemTemplate	DataTemplate
(description)	The DataTemplate used to display each item.

ItemsPanelTemplate

x:Object > DependencyObject > FrameworkTemplate > ItemsPanelTemplate

(usage)	<ItemsPanelTemplate>**FrameworkElement**</ItemsPanelTemplate>
(description)	Specifies the panel that the ItemsPresenter creates for the layout of the items of an ItemsControl.
(used by)	ItemsControl

(usage)	<ItemsPanelTemplate>**FrameworkElement**</ItemsPanelTemplate>
[content property]	Template

ItemsPresenter

x:Object > DependencyObject > UIElement > FrameworkElement > ItemsPresenter

(usage)	<ItemsPresenter />
(description)	Specifies where items are placed in a control, usually an ItemsControl.
[name property]	Name
[xml lang property]	Language

Key

x:Object > Key

(usage)	**None**| **Back**| **Tab**| **Enter**| **Shift**| **Ctrl**| **Alt**| **CapsLock**| **Escape**...
(description)	Specifies the possible key values on a keyboard.
[is nullable]	False
[text syntax]	KeySyntax

Keyboard

x:Object > Keyboard

(usage)	{x:Static Keyboard.StaticPropertyName}
(description)	Represents the keyboard device.
[is default constructible]	False
(static properties)	
Modifiers	ModifierKeys
(description)	The set of ModifierKeys that are currently pressed.

KeyboardNavigationMode

x:Object > KeyboardNavigationMode

(usage)	**Local**| **Cycle**| **Once**
(description)	Specifies the tabbing behavior across tab stops for a tabbing sequence within a container.

(usage)	**Local**| **Cycle**| **Once**
(used by)	Control
[is nullable]	False
[text syntax]	KeyboardNavigationModeSyntax

KeySpline

x:Object > DependencyObject > KeySpline

(usage)	<KeySpline> string </KeySpline>
(description)	This type is used by a spline key frame to define animation progress.
(used by)	SplineColorKeyFrame SplineDoubleKeyFrame SplinePointKeyFrame
[text syntax]	KeySplineSyntax
(properties)	
ControlPoint1	Point
(description)	The first control point used to define a Bezier curve that describes a KeySpline.
ControlPoint2	Point
(description)	The second control point used to define a Bezier curve that describes a KeySpline.

KeyTime

x:Object > KeyTime

(usage)	<KeyTime> string </KeyTime>
(description)	Specifies when a particular key frame should take place during an animation.
(used by)	ColorKeyFrame DoubleKeyFrame ObjectKeyFrame PointKeyFrame
[is nullable]	False
[text syntax]	KeyTimeSyntax
(static properties)	
Uniform	KeyTime
(description)	The Uniform value which divides the allotted time of the animation evenly between key frames.

KeyTimeType

x:Object > KeyTimeType

(usage)	**Uniform**| **TimeSpan**
(description)	Represents the different types that may represent a KeyTime instance.
[is nullable]	False
[text syntax]	KeyTimeTypeSyntax

LicenseAcquirer

x:Object > LicenseAcquirer

(usage)	<LicenseAcquirer />
(description)	This type handles acquiring licenses for DRM encrypted content.
(used by)	MediaElement
(properties)	
ChallengeCustomData (4)	x:String
(description)	A string containing service specific data to be conveyed to the license server without implementing manual license acquisition.
DomainAcquirer (4)	DomainAcquirer
(description)	A DomainAcquirer to handle Join Domain requests that are triggered from License Server exceptions (DomainRequired or RenewDomain).
LicenseServerUriOverride	x:Uri
(description)	A URI value that overrides whatever the license server URI is in the content header.
(events)	
AcquireLicenseCompleted (4)	Occurs when the license acquisition completes.

LicenseManagement (4)

x:Object > LicenseManagement

(usage)	None.
(description)	A static type that is used to return licenses from the persistent license store.
[is default constructible]	False

Line

x:Object > DependencyObject > UIElement > FrameworkElement > Shape > Line

(usage)	<Line />
(description)	Draws a straight line between two points.
[name property]	Name
[xml lang property]	Language
(properties)	
X1	x:Double
(description)	The x-coordinate of the Line start point.
X2	x:Double
(description)	The x-coordinate of the Line end point.
Y1	x:Double
(description)	The y-coordinate of the Line start point.
Y2	x:Double
(description)	The y-coordinate of the Line end point.

LinearColorKeyFrame

x:Object > DependencyObject > ColorKeyFrame > LinearColorKeyFrame

(usage)	<LinearColorKeyFrame />
(description)	Animates from the Color value of the previous key frame to its own Value using linear interpolation.

LinearDoubleKeyFrame

x:Object > DependencyObject > DoubleKeyFrame > LinearDoubleKeyFrame

(usage)	<LinearDoubleKeyFrame />
(description)	Animates from the Double value of the previous key frame to its own Value using linear interpolation.

LinearGradientBrush

x:Object > DependencyObject > Brush > GradientBrush > LinearGradientBrush

(usage)	<LinearGradientBrush>**GradientStop***</LinearGradientBrush>
(description)	Paints an area with a linear gradient.
[content property]	GradientStops
(properties)	

(usage)	<LinearGradientBrush>**GradientStop***</LinearGradientBrush>
EndPoint	Point
(description)	The ending two-dimensional coordinates of the linear gradient.
StartPoint	Point
(description)	The starting two-dimensional coordinates of the linear gradient.

LinearPointKeyFrame

x:Object > DependencyObject > PointKeyFrame > LinearPointKeyFrame

(usage)	<LinearPointKeyFrame />
(description)	Animates from the Point value of the previous key frame to its own Value using linear interpolation.

LineBreak

x:Object > DependencyObject > TextElement > Inline > LineBreak

(usage)	<LineBreak />
(description)	Represents an inline element that causes a new line to begin in content when rendered in a text container.
[xml lang property]	Language
[trim surrounding whitespace]	True

LineGeometry

x:Object > DependencyObject > Geometry > LineGeometry

(usage)	<LineGeometry />
(description)	Represents the geometry of a line.
(properties)	
EndPoint	Point
(description)	The end point of a line.
StartPoint	Point
(description)	The start point of the line.

LineSegment

x:Object > DependencyObject > PathSegment > LineSegment

(usage)	<LineSegment />
(description)	Represents a line drawn between two points, which can be part of a PathFigure within Path data.
(properties)	
Point	Point
(description)	The end point of the line segment.

LineStackingStrategy

x:Object > LineStackingStrategy

(usage)	**MaxHeight**| **BlockLineHeight**
(description)	Describes the mechanism by which a line box is determined for each line.
(used by)	TextBlock
[is nullable]	False
[text syntax]	LineStackingStrategySyntax

ListBox

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ItemsControl > Selector > ListBox

(usage)	<ListBox>**x:Object***</ListBox>
(description)	Contains a list of selectable items.
[content property]	Items
[name property]	Name
[xml lang property]	Language
(properties)	
ItemContainerStyle	Style
(description)	The style that is used when rendering the item containers.
SelectedItems	IList
(description)	The list of currently selected items for the ListBox control.
[read only]	True
SelectionMode	SelectionMode
(description)	The selection behavior for the ListBox control.

ListBoxItem

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ListBoxItem

ComboBoxItem	
(usage)	<ListBoxItem>x:Object</ListBoxItem>
(description)	Represents a selectable item in a ListBox.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
IsSelected	x:Boolean
(description)	A value that indicates whether a ListBoxItem is selected.

LogicalDirection (4)

x:Object > LogicalDirection

(usage)	**Backward**| **Forward**
(description)	Specifies a logical direction in which to perform certain text operations, such as inserting, retrieving, or navigating through text relative to a specified position (a TextPointer).
[is nullable]	False
[text syntax]	LogicalDirectionSyntax

LogSource

x:Object > LogSource

(usage)	**RequestLog**| **Stop**| **Seek**| **Pause**| **SourceChanged**| **EndOfStream**| **MediaElementShutdown**| **RuntimeShutdown**
(description)	Indicates the reason that a media log was generated.
[is nullable]	False
[text syntax]	LogSourceSyntax

Matrix

x:Object > Matrix

(usage)	<Matrix> string </Matrix>

(usage)	<Matrix> string </Matrix>
(description)	Represents a 3x3 affine transformation matrix used for transformations in two-dimensional space.
(used by)	MatrixTransform
[is nullable]	False
[text syntax]	MatrixSyntax
(properties)	
M11	x:Double
(description)	The value of the first row and first column of this Matrix structure.
M12	x:Double
(description)	The value of the first row and second column of this Matrix structure.
M21	x:Double
(description)	The value of the second row and first column of this Matrix structure.
M22	x:Double
(description)	The value of the second row and second column of this Matrix structure.
OffsetX	x:Double
(description)	The value of the third row and first column of this Matrix structure.
OffsetY	x:Double
(description)	The value of the third row and second column of this Matrix structure.
(static properties)	
Identity	Matrix
(description)	An identity Matrix.

Matrix3D

x:Object > Matrix3D

(usage)	<Matrix3D> string </Matrix3D>
(description)	Represents a 4 × 4 matrix that is used for transformations in a three-dimensional (3-D) space.
(used by)	Matrix3DProjection
[is nullable]	False
[text syntax]	Matrix3DSyntax

(usage)	<Matrix3D> string </Matrix3D>
(properties)	
M11	x:Double
(description)	The value of the first row and first column of this Matrix3D.
M12	x:Double
(description)	The value of the first row and second column of this Matrix3D.
M13	x:Double
(description)	The value of the first row and third column of this Matrix3D.
M14	x:Double
(description)	The value of the first row and fourth column of this Matrix3D.
M21	x:Double
(description)	The value of the second row and first column of this Matrix3D.
M22	x:Double
(description)	The value of the second row and second column of this Matrix3D.
M23	x:Double
(description)	The value of the second row and third column of this Matrix3D.
M24	x:Double
(description)	The value of the second row and fourth column of this Matrix3D.
M31	x:Double
(description)	The value of the third row and first column of this Matrix3D.
M32	x:Double
(description)	The value of the third row and second column of this Matrix3D.
M33	x:Double
(description)	The value of the third row and third column of this Matrix3D.
M34	x:Double
(description)	The value of the third row and fourth column of this Matrix3D.
M44	x:Double
(description)	The value of the fourth row and fourth column of this Matrix3D.
OffsetX	x:Double
(description)	The value of the fourth row and first column of this Matrix3D.
OffsetY	x:Double

(usage)	<Matrix3D> string </Matrix3D>
(description)	The value of the fourth row and second column of this Matrix3D.
OffsetZ	x:Double
(description)	The value of the fourth row and third column of this Matrix3D.
(static properties)	
Identity	Matrix3D
(description)	Changes a Matrix3D structure into an identity Matrix3D.

Matrix3DProjection

x:Object > DependencyObject > Projection > Matrix3DProjection

(usage)	<Matrix3DProjection>**Matrix3D**</Matrix3DProjection>
(description)	Enables you to apply a Matrix3D to an object.
[content property]	ProjectionMatrix
(properties)	
ProjectionMatrix	Matrix3D
(description)	The Matrix3D that is used for the projection that is applied to the object.

MatrixTransform

x:Object > DependencyObject > GeneralTransform > Transform > MatrixTransform

(usage)	<MatrixTransform> string </MatrixTransform>
(description)	Creates an arbitrary affine matrix transformation that is used to manipulate objects or coordinate systems in a two-dimensional plane.
[text syntax]	MatrixTransformSyntax
(properties)	
Matrix	Matrix
(description)	The Matrix that defines this transformation.

MediaElement

x:Object > DependencyObject > UIElement > FrameworkElement > MediaElement

(usage)	<MediaElement />
(description)	Represents an object that contains audio, video, or both.

(usage)	<MediaElement />
[name property]	Name
[xml lang property]	Language
(properties)	
Attributes	Dictionary(x:String,x:String)
(description)	The collection of attributes that corresponds to the current entry in the ASX file that Source is set to.
[read only]	True
AudioStreamIndex	x:Nullable(x:Int32)
(description)	The index of the audio stream that plays along with the video component. The collection of audio streams is composed at run time and represents all audio streams available within the media file.
AutoPlay	x:Boolean
(description)	A value that indicates whether media will begin playback automatically when the Source property is set.
Balance	x:Double
(description)	A ratio of volume across stereo speakers.
BufferingTime	x:TimeSpan
(description)	The amount of time to buffer.
IsMuted	x:Boolean
(description)	A value indicating whether the audio is muted.
LicenseAcquirer	LicenseAcquirer
(description)	The LicenseAcquirer associated with the MediaElement. The LicenseAcquirer handles acquiring licenses for DRM encrypted content.
Markers	TimelineMarkerCollection
(description)	The collection of timeline markers associated with the currently loaded media file.
[read only]	True
Position	x:TimeSpan
(description)	The current position of progress through the media's playback time.
Source	x:Uri
(description)	A media source on the MediaElement.
Stretch	Stretch

(usage)	<MediaElement />
(description)	A Stretch value that describes how a MediaElement fills the destination rectangle.
Volume	x:Double
(description)	The media's volume.
(events)	
BufferingProgressChanged	Occurs when the BufferingProgress property changes.
CurrentStateChanged	Occurs when the value of the CurrentState property changes.
DownloadProgressChanged	Occurs when the DownloadProgress property has changed.
LogReady	Occurs when the log is ready.
MarkerReached	Occurs when a timeline marker is encountered during media playback.
MediaEnded	Occurs when the MediaElement is no longer playing audio or video.
MediaFailed	Occurs when there is an error associated with the media Source.
MediaOpened	Occurs when the media stream has been validated and opened, and the file headers have been read.

MediaElementState

x:Object > MediaElementState

(usage)	**Closed**| **Opening**| **Individualizing**| **AcquiringLicense**| **Buffering**| **Playing**| **Paused**| **Stopped**
(description)	Defines the potential states of a MediaElement object.
[is nullable]	False
[text syntax]	MediaElementStateSyntax

MediaSampleAttributeKeys

x:Object > MediaSampleAttributeKeys

(usage)	**KeyFrameFlag**| **DRMInitializationVector**| **FrameWidth**| **FrameHeight**| **DRMSubSampleMapping**
(description)	Describes the media sample.
[is nullable]	False
[text syntax]	MediaSampleAttributeKeysSyntax

MediaSourceAttributesKeys

x:Object > MediaSourceAttributesKeys

(usage)	**CanSeek**| **Duration**| **DRMHeader**
(description)	Describes the media source.
[is nullable]	False
[text syntax]	MediaSourceAttributesKeysSyntax

MediaStreamAttributeKeys

x:Object > MediaStreamAttributeKeys

(usage)	**CodecPrivateData**| **VideoFourCC**| **Width**| **Height**
(description)	Specifies an attribute name that describes the media stream.
[is nullable]	False
[text syntax]	MediaStreamAttributeKeysSyntax

MediaStreamSourceDiagnosticKind

x:Object > MediaStreamSourceDiagnosticKind

(usage)	**BufferLevelInMilliseconds**| **BufferLevelInBytes**
(description)	Describes the type of diagnostic information used by the media.
[is nullable]	False
[text syntax]	MediaStreamSourceDiagnosticKindSyntax

MediaStreamType

x:Object > MediaStreamType

(usage)	**Audio**| **Video**| **Script**
(description)	Enumeration that specifies the type of stream.
[is nullable]	False
[text syntax]	MediaStreamTypeSyntax

MessageBoxButton

x:Object > MessageBoxButton

(usage)	**OK**| **OKCancel**

(usage)	**OK**| **OKCancel**
(description)	Specifies the buttons to include when you display a message box.
[is nullable]	False
[text syntax]	MessageBoxButtonSyntax

MessageBoxResult

x:Object > MessageBoxResult

(usage)	**None**| **OK**| **Cancel**| **Yes**| **No**
(description)	Represents a user's response to a message box.
[is nullable]	False
[text syntax]	MessageBoxResultSyntax

ModifierKeys

x:Object > ModifierKeys

(usage)	**None**| **Alt**| **Control**| **Shift**| **Windows**| **Apple**
(description)	Specifies the set of modifier keys.
[is nullable]	False
[text syntax]	ModifierKeysSyntax

MultiScaleImage

x:Object > DependencyObject > UIElement > FrameworkElement > MultiScaleImage

(usage)	<MultiScaleImage />
(description)	Enables users to open a multi-resolution image, which can be zoomed in on and panned across.
[name property]	Name
[xml lang property]	Language
(properties)	
AllowDownloading	x:Boolean
(description)	A value that indicates whether downloading is permitted by this MultiScaleImage.
BlurFactor	x:Double
(description)	The extent that data is blurred while rendering.

(usage)	<MultiScaleImage />
Source	MultiScaleTileSource
(description)	The MultiScaleTileSource object that is used as the source for the MultiScaleImage.
UseSprings	x:Boolean
(description)	A value that indicates whether the MultiScaleImage uses spring animations.
ViewportOrigin	Point
(description)	The top-left corner of the area of the image to be displayed.
ViewportWidth	x:Double
(description)	The width of the area of the image displayed.
(events)	
ImageFailed	Occurs if the download of a tile times out or fails for another reason.
ImageOpenFailed	Occurs if the first piece of metadata used to open the image fails. If this event occurs no parts of the image will open successfully.
ImageOpenSucceeded	Occurs when the first piece of metadata that is needed to load the rest of the tiles opens.
MotionFinished	Occurs when the zoom or pan animation ends.
ViewportChanged	Occurs when the viewport (the area of the image displayed) changes.

MultiScaleSubImage

x:Object > DependencyObject > MultiScaleSubImage

(usage)	<MultiScaleSubImage />
(description)	This type holds the properties for each sub-image within the MultiScaleImage.
(properties)	
Opacity	x:Double
(description)	The degree of the MultiScaleSubImage opacity.
ViewportOrigin	Point
(description)	The top-left corner of the area of the image to be displayed.
ViewportWidth	x:Double
(description)	The width of the area of the image displayed.
ZIndex	x:Int32
(description)	A value that represents the z-order rendering behavior of the MultiScaleSubImage. Z-order determines the relative rendering order of objects (which object is on top of

(usage)	<MultiScaleSubImage />
	which other objects).

MultiScaleTileSource

x:Object > DependencyObject > MultiScaleTileSource

DeepZoomImageTileSource	
(usage)	<MultiScaleTileSource> string </MultiScaleTileSource>
(description)	Used to specify the source of Deep Zoom images.
(used by)	MultiScaleImage
[is default constructible]	False
[text syntax]	x:Uri, from [MS-XAML]

NotificationWindow (4)

x:Object > DependencyObject > NotificationWindow

(usage)	<NotificationWindow />
(description)	Represents a notification area that is displayed in the system area. Notifications can only be enabled for an out-of-browser application; browser-hosted applications cannot access this notification area.
(properties)	
Content	FrameworkElement
(description)	The root of visual elements that define the visual look of the notification.
Height	x:Double
(description)	The height, in pixels, of this notification window. See Remarks.
Width	x:Double
(description)	The width, in pixels, of this notification window. See Remarks.
(events)	
Closed	Occurs when Close is called, or when the notification window times out and has finished its fadeout animation.

ObjectAnimationUsingKeyFrames

x:Object > DependencyObject > Timeline > ObjectAnimationUsingKeyFrames

(usage)	<ObjectAnimationUsingKeyFrames>**ObjectKeyFrame***</ObjectAnimationUsingKeyFrames>
(description)	Animates the value of an Object property along a set of KeyFrames over a specified

(usage)	<ObjectAnimationUsingKeyFrames>**ObjectKeyFrame***</ObjectAnimationUsingKeyFrames>
	Duration.
[content property]	KeyFrames
(properties)	
KeyFrames	ObjectKeyFrameCollection
(description)	The collection of ObjectKeyFrame objects that define the animation.
[read only]	True

ObjectKeyFrame

x:Object > DependencyObject > ObjectKeyFrame

DiscreteObjectKeyFrame	
(usage)	None.
(description)	Defines an animation segment with its own target value and interpolation technique for an ObjectAnimationUsingKeyFrames.
(used by)	ObjectKeyFrameCollection
[is default constructible]	False
(properties)	
KeyTime	KeyTime
(description)	The time at which the key frame's target Value should be reached.
Value	x:Object
(description)	The key frame's target value.

ObjectKeyFrameCollection

x:Object > DependencyObject > PresentationFrameworkCollection(ObjectKeyFrame) > ObjectKeyFrameCollection

(usage)	<ObjectKeyFrameCollection>**ObjectKeyFrame***</ObjectKeyFrameCollection>
(description)	Represents a collection of ObjectKeyFrame objects that can be individually accessed by index.
(used by)	ObjectAnimationUsingKeyFrames
[is list]	True
[allowed types]	ObjectKeyFrame

OpenFileDialog

x:Object > OpenFileDialog

(usage)	<OpenFileDialog />
(description)	Provides a dialog box that enables the user to select one or more files.
(properties)	
Filter	x:String
(description)	A filter string that specifies the file types and descriptions to display in the OpenFileDialog.
FilterIndex	x:Int32
(description)	The index of the selected item in the OpenFileDialog filter drop-down list.
Multiselect	x:Boolean
(description)	A value that indicates whether the OpenFileDialog allows users to select multiple files.

Orientation

x:Object > Orientation

(usage)	**Vertical**| **Horizontal**
(description)	Defines the different orientations that a control or layout can have.
(used by)	ScrollBar Slider StackPanel VirtualizingStackPanel
[is nullable]	False
[text syntax]	OrientationSyntax

OutOfBrowserSettings

x:Object > DependencyObject > OutOfBrowserSettings

(usage)	<OutOfBrowserSettings />
(description)	Represents information about an application that is configured for out-of-browser support.
(properties)	
Icons	IconCollection
(description)	A collection of Icon instances associated with the application.
[read only]	True

Panel

x:Object > DependencyObject > UIElement > FrameworkElement > Panel

Canvas Grid StackPanel VirtualizingPanel	
(usage)	None.
(description)	Provides a base typefor all Panel elements. Use Panel elements to position and arrange child objects.
[is default constructible]	False
[content property]	Children
[name property]	Name
[xml lang property]	Language
(properties)	
Background	Brush
(description)	A Brush that is used to fill the panel.
Children	UIElementCollection
(description)	The collection of child elements of the panel.
[read only]	True

Paragraph (4)

x:Object > DependencyObject > TextElement > Block > Paragraph

(usage)	<Paragraph>[**Inline**|**x:String**|**UIElement**]*</Paragraph>
(description)	Provides a block-level content element that is used to group content into a paragraph.
[content property]	Inlines
(properties)	
Inlines	InlineCollection
(description)	An InlineCollection containing the top-level Inline elements that include the contents of the Paragraph.
[read only]	True

PasswordBox

x:Object > DependencyObject > UIElement > FrameworkElement > Control > PasswordBox

(usage)	<PasswordBox />
(description)	Represents a control for entering passwords.

(usage)	<PasswordBox />
[name property]	Name
[xml lang property]	Language
(properties)	
CaretBrush	Brush
(description)	The brush that is used to render the vertical bar that indicates the insertion point.
MaxLength	x:Int32
(description)	The maximum length for passwords to be handled by this PasswordBox.
Password	x:String
(description)	The password currently held by the PasswordBox.
PasswordChar	x:Char
(description)	The masking character for the PasswordBox.
SelectionBackground	Brush
(description)	The brush used to render the background for the selected text.
SelectionForeground	Brush
(description)	The brush used for the selected text in the PasswordBox.
(events)	
PasswordChanged	Occurs when the value of the Password property changes.

Path

x:Object > DependencyObject > UIElement > FrameworkElement > Shape > Path

(usage)	<Path />
(description)	Draws a series of connected lines and curves. The line and curve dimensions are declared through the Data property, and can be specified either with a Path-specific mini-language, or with an object model.
[name property]	Name
[xml lang property]	Language
(properties)	
Data	Geometry
(description)	A Geometry that specifies the shape to be drawn.

PathFigure

x:Object > DependencyObject > PathFigure

(usage)	<PathFigure>**PathSegment***</PathFigure>
(description)	Represents a subsection of a geometry, a single connected series of two-dimensional geometric segments.
(used by)	PathFigureCollection
[content property]	Segments
(properties)	
IsClosed	x:Boolean
(description)	A value that indicates whether this figure's first and last segments are connected.
IsFilled	x:Boolean
(description)	A value that indicates whether the contained area of this PathFigure is to be used for hit-testing, rendering, and clipping.
Segments	PathSegmentCollection
(description)	The collection of segments that define the shape of this PathFigure object.
StartPoint	Point
(description)	The Point where the PathFigure begins.

PathFigureCollection

x:Object > DependencyObject > PresentationFrameworkCollection(PathFigure) > PathFigureCollection

(usage)	<PathFigureCollection>**PathFigure***</PathFigureCollection>
(description)	Represents a collection of PathFigure objects that collectively make up the geometry of a PathGeometry.
(used by)	PathGeometry
[is list]	True
[allowed types]	PathFigure

PathGeometry

x:Object > DependencyObject > Geometry > PathGeometry

(usage)	<PathGeometry>**PathFigure***</PathGeometry>
(description)	Represents a complex shape that may be composed of arcs, curves, ellipses, lines,

(usage)	<PathGeometry>**PathFigure***</PathGeometry>
	and rectangles.
[content property]	Figures
(properties)	
Figures	PathFigureCollection
(description)	The collection of PathFigure objects that describe the contents of a path.
FillRule	FillRule
(description)	A value that determines how the intersecting areas contained in the PathGeometry are combined.

PathSegment

x:Object > DependencyObject > PathSegment

ArcSegment BezierSegment LineSegment PolyBezierSegment PolyLineSegment PolyQuadraticBezierSegment QuadraticBezierSegment	
(usage)	None.
(description)	Represents a segment of a PathFigure object.
(used by)	PathSegmentCollection
[is default constructible]	False

PathSegmentCollection

x:Object > DependencyObject > PresentationFrameworkCollection(PathSegment) > PathSegmentCollection

(usage)	<PathSegmentCollection>**PathSegment***</PathSegmentCollection>
(description)	Represents a collection of PathSegment objects that can be individually accessed by index.
(used by)	PathFigure
[is list]	True
[allowed types]	PathSegment

PenLineCap

x:Object > PenLineCap

(usage)	**Flat**| **Square**| **Round**| **Triangle**

(usage)	**Flat**| **Square**| **Round**| **Triangle**
(description)	Describes the shape at the end of a line or segment.
(used by)	Shape
[is nullable]	False
[text syntax]	PenLineCapSyntax

PenLineJoin

x:Object > PenLineJoin

(usage)	**Miter**| **Bevel**| **Round**
(description)	Describes the shape that joins two lines or segments.
(used by)	Shape
[is nullable]	False
[text syntax]	PenLineJoinSyntax

PixelFormatType (4)

x:Object > PixelFormatType

(usage)	**Unknown**| **Format32bppArgb**
(description)	Describes format information that reports the graphics format of individual pixels of the video format.
[is nullable]	False
[text syntax]	PixelFormatTypeSyntax

PixelShader

x:Object > DependencyObject > PixelShader

(usage)	<PixelShader> string </PixelShader>
(description)	Provides a managed wrapper around a High Level Shading Language (HLSL) pixel shader.
[text syntax]	PixelShaderSyntax
(properties)	
UriSource	x:Uri
(description)	A URI reference to HLSL bytecode in the assembly.

PlacementMode

x:Object > PlacementMode

(usage)	**Bottom**| **Right**| **Mouse**| **Left**| **Top**
(description)	Specifies the preferred location for positioning a ToolTip relative to a visual element.
(used by)	ToolTip ToolTipService
[is nullable]	False
[text syntax]	PlacementModeSyntax

PlaneProjection

x:Object > DependencyObject > Projection > PlaneProjection

(usage)	<PlaneProjection />
(description)	Represents a perspective transform (a 3-D-like effect) on an object.
(properties)	
CenterOfRotationX	x:Double
(description)	The x-coordinate of the center of rotation of the object you rotate.
CenterOfRotationY	x:Double
(description)	The y-coordinate of the center of rotation of the object you rotate.
CenterOfRotationZ	x:Double
(description)	The z-coordinate of the center of rotation of the object you rotate.
GlobalOffsetX	x:Double
(description)	The distance the object is translated along the x-axis of the screen.
GlobalOffsetY	x:Double
(description)	The distance the object is translated along the y-axis of the screen.
GlobalOffsetZ	x:Double
(description)	The distance the object is translated along the z-axis of the screen.
LocalOffsetX	x:Double
(description)	The distance the object is translated along the x-axis of the plane of the object.
LocalOffsetY	x:Double
(description)	The distance the object is translated along the y-axis of the plane of the object.
LocalOffsetZ	x:Double
(description)	The distance the object is translated along the z-axis of the plane of the object.

(usage)	<PlaneProjection />
RotationX	x:Double
(description)	The number of degrees to rotate the object around the x-axis of rotation.
RotationY	x:Double
(description)	The number of degrees to rotate the object around the y-axis of rotation.
RotationZ	x:Double
(description)	The number of degrees to rotate the object around the z-axis of rotation.

Point

x:Object > Point

(usage)	<Point> string </Point>
(description)	Represents an x- and y-coordinate pair in two-dimensional space. Can also represent a logical point for certain property usages.
(used by)	ArcSegment BezierSegment EllipseGeometry KeySpline LinearGradientBrush LineGeometry LineSegment MultiScaleImage MultiScaleSubImage PathFigure PointCollection PointKeyFrame QuadraticBezierSegment RadialGradientBrush UIElement
[is nullable]	False
[text syntax]	PointSyntax
(properties)	
X	x:Double
(description)	The X-coordinate value of this Point structure.
Y	x:Double
(description)	The Y-coordinate value of this Point.

PointAnimation

x:Object > DependencyObject > Timeline > PointAnimation

(usage)	<PointAnimation />
(description)	Animates the value of a Point property between two target values using linear interpolation over a specified Duration.
(properties)	
By	x:Nullable(Point)
(description)	The total amount by which the animation changes its starting value.

(usage)	<PointAnimation />
EasingFunction	IEasingFunction
(description)	The easing function you are applying to the animation.
From	x:Nullable(Point)
(description)	The animation's starting value.
To	x:Nullable(Point)
(description)	The animation's ending value.

PointAnimationUsingKeyFrames

x:Object > DependencyObject > Timeline > PointAnimationUsingKeyFrames

(usage)	<PointAnimationUsingKeyFrames>**PointKeyFrame***</PointAnimationUsingKeyFrames>
(description)	Animates the value of a Point property along a set of KeyFrames.
[content property]	KeyFrames
(properties)	
KeyFrames	PointKeyFrameCollection
(description)	The collection of PointKeyFrame objects that define the animation.
[read only]	True

PointCollection

x:Object > DependencyObject > PresentationFrameworkCollection(Point) > PointCollection

(usage)	<PointCollection> string </PointCollection>
(description)	Represents a collection of Point values that can be individually accessed by index.
(used by)	PolyBezierSegment Polygon Polyline PolyLineSegment PolyQuadraticBezierSegment
[text syntax]	PointCollectionSyntax
[is list]	True
[allowed types]	Point

PointKeyFrame

x:Object > DependencyObject > PointKeyFrame

DiscretePointKeyFrame EasingPointKeyFrame LinearPointKeyFrame SplinePointKeyFrame	
(usage)	None.

DiscretePointKeyFrame EasingPointKeyFrame LinearPointKeyFrame SplinePointKeyFrame	
(description)	Defines an animation segment with its own target value and interpolation technique for a PointAnimationUsingKeyFrames.
(used by)	PointKeyFrameCollection
[is default constructible]	False
(properties)	
KeyTime	KeyTime
(description)	The time at which the key frame's target Value should be reached.
Value	Point
(description)	The key frame's target value.

PointKeyFrameCollection

x:Object > DependencyObject > PresentationFrameworkCollection(PointKeyFrame) > PointKeyFrameCollection

(usage)	<PointKeyFrameCollection>**PointKeyFrame***</PointKeyFrameCollection>
(description)	Represents a collection of PointKeyFrame objects that can be individually accessed by index.
(used by)	PointAnimationUsingKeyFrames
[is list]	True
[allowed types]	PointKeyFrame

PolyBezierSegment

x:Object > DependencyObject > PathSegment > PolyBezierSegment

(usage)	<PolyBezierSegment>**Point***</PolyBezierSegment>
(description)	Represents one or more cubic Bezier curves.
[content property]	Points
(properties)	
Points	PointCollection
(description)	The PointCollection that defines this PolyBezierSegment object.

Polygon

x:Object > DependencyObject > UIElement > FrameworkElement > Shape > Polygon

(usage)	<Polygon />
(description)	Draws a polygon, which is a connected series of lines that form a closed shape.
[name property]	Name
[xml lang property]	Language
(properties)	
FillRule	FillRule
(description)	A value that specifies how the interior fill of the shape is determined.
Points	PointCollection
(description)	A collection that contains the vertex points of the polygon.

Polyline

x:Object > DependencyObject > UIElement > FrameworkElement > Shape > Polyline

(usage)	<Polyline />
(description)	Draws a series of connected straight lines.
[name property]	Name
[xml lang property]	Language
(properties)	
FillRule	FillRule
(description)	A value that specifies how the interior fill of the shape is determined.
Points	PointCollection
(description)	A collection that contains the vertex points of the Polyline.

PolyLineSegment

x:Object > DependencyObject > PathSegment > PolyLineSegment

(usage)	<PolyLineSegment>**Point***</PolyLineSegment>
(description)	Represents a set of line segments defined by a PointCollection with each Point specifying the end point of a line segment.
[content property]	Points
(properties)	
Points	PointCollection
(description)	The collection of Point values that defines this PolyLineSegment object.

PolyQuadraticBezierSegment

x:Object > DependencyObject > PathSegment > PolyQuadraticBezierSegment

(usage)	<PolyQuadraticBezierSegment>**Point***</PolyQuadraticBezierSegment>
(description)	Represents a set of quadratic Bezier segments.
[content property]	Points
(properties)	
Points	PointCollection
(description)	The PointCollection that defines this PolyQuadraticBezierSegment object.

Popup

x:Object > DependencyObject > UIElement > FrameworkElement > Popup

(usage)	<Popup>**UIElement**</Popup>
(description)	Displays content on top of existing content, within the bounds of the hosting content.
[content property]	Child
[name property]	Name
[xml lang property]	Language
(properties)	
Child	UIElement
(description)	The content to be hosted in the popup.
HorizontalOffset	x:Double
(description)	The distance between the left side of the hosting content and the left side of the popup.
IsOpen	x:Boolean
(description)	Whether the popup is currently displaying on the screen.
VerticalOffset	x:Double
(description)	The distance between the top of the hosting content and the top of the popup.
(events)	
Closed	Occurs when the IsOpen property is set to false.
Opened	Occurs when the IsOpen property is set to true.

PowerEase

x:Object > DependencyObject > EasingFunctionBase > PowerEase, IEasingFunction

(usage)	<PowerEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using the formula f(t) = tp where p is equal to the Power property.
(properties)	
Power	x:Double
(description)	The exponential power of the animation interpolation. For example, a value of 7 creates an animation interpolation curve that follows the formula f(t) = t7.

PresentationFrameworkCollection(T)

x:Object > DependencyObject > PresentationFrameworkCollection(T)

(usage)	None.
(description)	Provides a common base type for many collections.
[is default constructible]	False
[is list]	True
[allowed types]	T

ProgressBar

x:Object > DependencyObject > UIElement > FrameworkElement > Control > RangeBase > ProgressBar

(usage)	<ProgressBar />
(description)	Represents a control that indicates the progress of an operation.
[name property]	Name
[xml lang property]	Language
(properties)	
IsIndeterminate	x:Boolean
(description)	A value that indicates whether the progress bar reports generic progress with a repeating pattern or reports progress based on the Value property.

Projection

x:Object > DependencyObject > Projection

Matrix3DProjection PlaneProjection

Matrix3DProjection PlaneProjection	
(usage)	None.
(description)	Provides a base type for projections, which describe how to transform an object in 3-D space using perspective transforms.
(used by)	UIElement
[is default constructible]	False

PropertyGroupDescription (4)

x:Object > GroupDescription > PropertyGroupDescription

(usage)	<PropertyGroupDescription />
(description)	Describes the grouping of items by using a property name as the criteria.
(properties)	
PropertyName	x:String
(description)	The name of the property that is used to determine which group(s) an item belongs to.
StringComparison	StringComparison
(description)	A StringComparison value that specifies the comparison between the value of an item (as determined by PropertyName and Converter) and the name of a group.

PropertyPath

x:Object > PropertyPath

(usage)	<PropertyPath> string </PropertyPath>
(description)	Implements a data structure for describing a property as a path below another property, or below an owning type. Property paths are used in data binding to objects, and in storyboards and timelines for animations.
(used by)	Binding Storyboard
[is default constructible]	False
[text syntax]	PropertyPathSyntax
(properties)	
Path	x:String
(description)	The path value held by this PropertyPath.

QuadraticBezierSegment

x:Object > DependencyObject > PathSegment > QuadraticBezierSegment

(usage)	<QuadraticBezierSegment />
(description)	Creates a quadratic Bezier curve between two points in a PathFigure.
(properties)	
Point1	Point
(description)	The control point of the curve.
Point2	Point
(description)	The end Point of this QuadraticBezierSegment.

QuadraticEase

x:Object > DependencyObject > EasingFunctionBase > QuadraticEase, IEasingFunction

(usage)	<QuadraticEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using the formula f(t) = t2

QuarticEase

x:Object > DependencyObject > EasingFunctionBase > QuarticEase, IEasingFunction

(usage)	<QuarticEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using the formula f(t) = t4.

QuinticEase

x:Object > DependencyObject > EasingFunctionBase > QuinticEase, IEasingFunction

(usage)	<QuinticEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using the formula f(t) = t5.

RadialGradientBrush

x:Object > DependencyObject > Brush > GradientBrush > RadialGradientBrush

(usage)	<RadialGradientBrush>**GradientStop***</RadialGradientBrush>
(description)	Paints an area with a radial gradient. A focal point defines the beginning of the gradient, and a circle defines the end point of the gradient.

(usage)	<RadialGradientBrush>**GradientStop***</RadialGradientBrush>
[content property]	GradientStops
(properties)	
Center	Point
(description)	The center of the outer circle of the radial gradient.
GradientOrigin	Point
(description)	The location of the focal point that defines the beginning of the gradient.
RadiusX	x:Double
(description)	The horizontal radius of the outer circle of the radial gradient.
RadiusY	x:Double
(description)	The vertical radius of the outer circle of a radial gradient.

RadioButton

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase > ToggleButton > RadioButton

(usage)	<RadioButton>**x:Object**</RadioButton>
(description)	Represents a button that allows a user to select a single option from a group of options.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
GroupName	x:String
(description)	The name that specifies which RadioButton controls are mutually exclusive.

RangeBase

x:Object > DependencyObject > UIElement > FrameworkElement > Control > RangeBase

ProgressBar ScrollBar Slider	
(usage)	None.
(description)	Represents an element that has a value within a specific range, such as the ProgressBar, ScrollBar, and Slider controls.

ProgressBar ScrollBar Slider	
[is default constructible]	False
[name property]	Name
[xml lang property]	Language
(properties)	
LargeChange	x:Double
(description)	A value to be added to or subtracted from the Value of a RangeBase control.
Maximum	x:Double
(description)	The highest possible Value of the range element.
Minimum	x:Double
(description)	The Minimum possible Value of the range element.
SmallChange	x:Double
(description)	A Value to be added to or subtracted from the Value of a RangeBase control.
Value	x:Double
(description)	The current setting of the range control, which may be coerced.
(events)	
ValueChanged	Occurs when the range value changes.

Rect

x:Object > Rect

(usage)	<Rect> string </Rect>
(description)	Describes the width, height, and point origin of a rectangle.
(used by)	RectangleGeometry
[is nullable]	False
[text syntax]	RectSyntax
(properties)	
Height	x:Double
(description)	The height of the rectangle.
Width	x:Double
(description)	The width of the rectangle.

(usage)	<Rect> string </Rect>
X	x:Double
(description)	The x-axis value of the left side of the rectangle.
Y	x:Double
(description)	The y-axis value of the top side of the rectangle.
(static properties)	
Empty	Rect
(description)	A special value that represents a rectangle with no position or area.

Rectangle

x:Object > DependencyObject > UIElement > FrameworkElement > Shape > Rectangle

(usage)	<Rectangle />
(description)	Draws a rectangle shape, which can have a stroke and a fill.
[name property]	Name
[xml lang property]	Language
(properties)	
RadiusX	x:Double
(description)	The x-axis radius of the ellipse that is used to round the corners of the rectangle.
RadiusY	x:Double
(description)	The y-axis radius of the ellipse that is used to round the corners of the rectangle.

RectangleGeometry

x:Object > DependencyObject > Geometry > RectangleGeometry

(usage)	<RectangleGeometry />
(description)	Describes a two-dimensional rectangular geometry.
(properties)	
RadiusX	x:Double
(description)	The x-radius of the ellipse that is used to round the corners of the rectangle.
RadiusY	x:Double
(description)	The y-radius of the ellipse that is used to round the corners of the rectangle.
Rect	Rect

(usage)	<RectangleGeometry />
(description)	The dimensions of the rectangle.

RelativeSource

x:Object > x:MarkupExtension > RelativeSource

(usage)	{RelativeSource } | <RelativeSource />
(description)	Implements a markup extension that describes the location of the binding source relative to the position of the binding target.
(used by)	Binding
[return value type]	RelativeSource
[constructors]	
(1 parameter)	
mode	RelativeSourceMode
(description)	The relative source mode to use for the related binding.
(properties)	
Mode	RelativeSourceMode
(description)	A value that describes the location of the binding source relative to the position of the binding target.

RelativeSourceMode

x:Object > RelativeSourceMode

(usage)	**TemplatedParent**| **Self**
(description)	Defines constants that describe the location of the binding source relative to the position of the binding target.
(used by)	RelativeSource
[is nullable]	False
[text syntax]	RelativeSourceModeSyntax

RepeatBehavior

x:Object > RepeatBehavior

(usage)	<RepeatBehavior> string </RepeatBehavior>
(description)	Describes how a Timeline repeats its simple duration.

(usage)	<RepeatBehavior> string </RepeatBehavior>
(used by)	Timeline
[is nullable]	False
[text syntax]	RepeatBehaviorSyntax
(static properties)	
Forever	RepeatBehavior
(description)	A RepeatBehavior that specifies an infinite number of repetitions.

RepeatButton

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase > RepeatButton

(usage)	<RepeatButton>**x:Object**</RepeatButton>
(description)	Represents a control that raises its Click event repeatedly from the time it is pressed until it is released.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
Delay	x:Int32
(description)	The time, in milliseconds, the RepeatButton waits when it is pressed before it starts repeating the click action.
Interval	x:Int32
(description)	The time, in milliseconds, between repetitions of the click action, as soon as repeating starts.

ResourceDictionary

x:Object > DependencyObject > ResourceDictionary

(usage)	<ResourceDictionary>**ResourceDictionary**</ResourceDictionary>
(description)	Provides a dictionary that contains keyed resources used by components of a Silverlight-based application.
(used by)	Application FrameworkElement ResourceDictionary
[is dictionary]	True

(usage)	<ResourceDictionary>**ResourceDictionary**</ResourceDictionary>
[allowed types]	x:Object
[allowed key types]	x:Object
(properties)	
MergedDictionaries	PresentationFrameworkCollection(ResourceDictionary)
(description)	A collection of the ResourceDictionary dictionaries that constitute the various resource dictionaries in the merged dictionaries.
[read only]	True
Source	x:Uri
(description)	A URI that provides the source location of a merged resource dictionary.

RichTextBox (4)

x:Object > DependencyObject > UIElement > FrameworkElement > Control > RichTextBox

(usage)	<RichTextBox>**Block***</RichTextBox>
(description)	Represents a rich text editing control that supports formatted text, hyperlinks, inline images, and other rich content.
[content property]	Blocks
[name property]	Name
[xml lang property]	Language
(properties)	
AcceptsReturn	x:Boolean
(description)	A value that determines whether the RichTextBox allows and displays the newline or return characters.
Blocks	BlockCollection
(description)	The contents of the RichTextBox.
[read only]	True
CaretBrush	Brush
(description)	The brush that is used to render the vertical bar that indicates the insertion point.
HorizontalScrollBarVisibility	ScrollBarVisibility
(description)	The visibility of the horizontal scroll bar.
IsReadOnly	x:Boolean

(usage)	<RichTextBox>**Block***</RichTextBox>
(description)	A value that determines whether the user can change the text in the RichTextBox.
TextAlignment	TextAlignment
(description)	How the text should be aligned in the text box.
TextWrapping	TextWrapping
(description)	How text wrapping occurs if a line of text extends beyond the available width of the RichTextBox.
VerticalScrollBarVisibility	ScrollBarVisibility
(description)	The visibility of the vertical scroll bar.
Xaml	x:String
(description)	A XAML representation of the content in the RichTextBox.
(events)	
ContentChanged	Occurs when the content changes in a RichTextBox.
SelectionChanged	Occurs when the text selection has changed.

RotateTransform

x:Object > DependencyObject > GeneralTransform > Transform > RotateTransform

(usage)	<RotateTransform />
(description)	Rotates an object clockwise about a specified point in a two-dimensional x-y coordinate system.
(properties)	
Angle	x:Double
(description)	The angle, in degrees, of clockwise rotation.
CenterX	x:Double
(description)	The x-coordinate of the rotation center point.
CenterY	x:Double
(description)	The y-coordinate of the rotation center point.

RowDefinition

x:Object > DependencyObject > RowDefinition

(usage)	<RowDefinition />

(usage)	<RowDefinition />
(description)	Defines row-specific properties that apply to Grid elements.
(used by)	RowDefinitionCollection
(properties)	
Height	GridLength
(description)	A value that represents the height of a RowDefinition.
MaxHeight	x:Double
(description)	A value that represents the maximum height of a RowDefinition.
MinHeight	x:Double
(description)	A value that represents the minimum allowed height of a RowDefinition.

RowDefinitionCollection

x:Object > DependencyObject > PresentationFrameworkCollection(RowDefinition) > RowDefinitionCollection

(usage)	None.
(description)	Provides access to an ordered, strongly typed collection of RowDefinition objects.
(used by)	Grid
[is default constructible]	False
[is list]	True
[allowed types]	RowDefinition

RowOrColumnMajor

x:Object > RowOrColumnMajor

(usage)	**RowMajor**| **ColumnMajor**| **Indeterminate**
(description)	Specifies whether data in a table should be read primarily by row or by column.
[is nullable]	False
[text syntax]	RowOrColumnMajorSyntax

Run

x:Object > DependencyObject > TextElement > Inline > Run

(usage)	<Run>**x:String**</Run>

(usage)	<Run>**x:String**</Run>
(description)	Represents a discrete section of formatted or unformatted text.
[content property]	Text
[xml lang property]	Language
(properties)	
FlowDirection (4)	FlowDirection
(description)	The direction that text and other user interface elements flow within the Run element that controls their layout.
Text	x:String
(description)	The text contents of the Run.

SamplingMode

x:Object > SamplingMode

(usage)	**Auto**| **NearestNeighbor**| **Bilinear**
(description)	Specifies how properties with Brush values are sampled in a custom shader effect.
[is nullable]	False
[text syntax]	SamplingModeSyntax

SaveFileDialog

x:Object > SaveFileDialog

(usage)	<SaveFileDialog />
(description)	Provides a dialog box that enables the user to specify options for saving a file.
(properties)	
DefaultExt	x:String
(description)	The default file name extension applied to files that are saved with the SaveFileDialog.
Filter	x:String
(description)	A filter string that specifies the files types and descriptions to display in the SaveFileDialog.
FilterIndex	x:Int32
(description)	The index of the selected item in the Save as type drop-down list.

ScaleTransform

x:Object > DependencyObject > GeneralTransform > Transform > ScaleTransform

(usage)	<ScaleTransform />
(description)	Scales an object in the two-dimensional x-y coordinate system.
(properties)	
CenterX	x:Double
(description)	The x-coordinate of the center point of this ScaleTransform.
CenterY	x:Double
(description)	The y-coordinate of the center point of this ScaleTransform.
ScaleX	x:Double
(description)	The x-axis scale factor.
ScaleY	x:Double
(description)	The y-axis scale factor.

ScrollAmount

x:Object > ScrollAmount

(usage)	**LargeDecrement**| **SmallDecrement**| **NoAmount**| **LargeIncrement**| **SmallIncrement**
(description)	Contains values that are used by the IScrollProvider pattern to indicate the direction and distance to scroll.
[is nullable]	False
[text syntax]	ScrollAmountSyntax

ScrollBar

x:Object > DependencyObject > UIElement > FrameworkElement > Control > RangeBase > ScrollBar

(usage)	<ScrollBar />
(description)	Represents a control that provides a scroll bar that has a sliding Thumb whose position corresponds to a value.
[name property]	Name
[xml lang property]	Language
(properties)	

(usage)	<ScrollBar />
Orientation	Orientation
(description)	Whether the ScrollBar is displayed horizontally or vertically.
ViewportSize	x:Double
(description)	The amount of the scrollable content that is currently visible.
(events)	
Scroll	Occurs one or more times as content scrolls in a ScrollBar when the user moves the Thumb by using the mouse.

ScrollBarVisibility

x:Object > ScrollBarVisibility

(usage)	**Disabled**| **Auto**| **Hidden**| **Visible**
(description)	Specifies the visibility of a scrollbar within a ScrollViewer control.
(used by)	RichTextBox ScrollViewer TextBox
[is nullable]	False
[text syntax]	ScrollBarVisibilitySyntax

ScrollContentPresenter

x:Object > DependencyObject > UIElement > FrameworkElement > ContentPresenter > ScrollContentPresenter

(usage)	<ScrollContentPresenter>**x:Object**</ScrollContentPresenter>
(description)	Displays the content of a ScrollViewer control.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
CanHorizontallyScroll	x:Boolean
(description)	A value that indicates whether scrolling on the horizontal axis is possible.
CanVerticallyScroll	x:Boolean
(description)	A value that indicates whether scrolling on the vertical axis is possible.
ScrollOwner	ScrollViewer
(description)	The ScrollViewer element that controls scrolling behavior.

ScrollEventType

x:Object > ScrollEventType

(usage)	**EndScroll**| **First**| **LargeDecrement**| **LargeIncrement**| **Last**| **SmallDecrement**| **SmallIncrement**| **ThumbPosition**| **ThumbTrack**...
(description)	Specifies the type of Scroll event that occurred.
[is nullable]	False
[text syntax]	ScrollEventTypeSyntax

ScrollViewer

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ScrollViewer

(usage)	<ScrollViewer>**x:Object**</ScrollViewer>
(description)	Represents a scrollable area that can contain other visible elements.
(used by)	IScrollInfo ScrollContentPresenter VirtualizingStackPanel
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
ComputedHorizontalScrollBarVisibility	Visibility
(description)	A value that indicates whether the horizontal ScrollBar is visible.
ComputedVerticalScrollBarVisibility	Visibility
(description)	A value that indicates whether the vertical ScrollBar is visible.
ExtentHeight	x:Double
(description)	The vertical size of all the content for display in the ScrollViewer.
ExtentWidth	x:Double
(description)	The horizontal size of all the content for display in the ScrollViewer.
HorizontalOffset	x:Double
(description)	A value that contains the horizontal offset of the scrolled content.

(usage)	<ScrollViewer>**x:Object**</ScrollViewer>
HorizontalScrollBarVisibility	ScrollBarVisibility
(description)	A value that indicates whether a horizontal ScrollBar should be displayed.
ScrollableHeight	x:Double
(description)	A value that represents the vertical size of the area that can be scrolled; the difference between the width of the extent and the width of the viewport.
ScrollableWidth	x:Double
(description)	A value that represents the horizontal size of the area that can be scrolled; the difference between the width of the extent and the width of the viewport..
VerticalOffset	x:Double
(description)	A value that contains the vertical offset of the scrolled content.
VerticalScrollBarVisibility	ScrollBarVisibility
(description)	A value that indicates whether a vertical ScrollBar should be displayed.
ViewportHeight	x:Double
(description)	A value that contains the vertical size of the viewable content.
ViewportWidth	x:Double
(description)	A value that contains the horizontal size of the viewable content.
(attachable properties)	
ScrollViewer.HorizontalScrollBarVisibility	ScrollBarVisibility
(description)	A value that indicates whether a horizontal ScrollBar should be displayed.
[target type]	DependencyObject
ScrollViewer.VerticalScrollBarVisibility	ScrollBarVisibility
(description)	A value that indicates whether a vertical ScrollBar should be displayed.
[target type]	DependencyObject

SecuritySettings (4)

x:Object > DependencyObject > SecuritySettings

(usage)	<SecuritySettings />
(description)	Represents the security configuration of an out-of-browser application.

SelectionMode

x:Object > SelectionMode

(usage)	**Single**| **Multiple**| **Extended**
(description)	Defines the selection behavior for a ListBox.
(used by)	ListBox
[is nullable]	False
[text syntax]	SelectionModeSyntax

Selector

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ItemsControl > Selector

ComboBox ListBox	
(usage)	None.
(description)	Represents a control that allows a user to select an item from a collection of items.
[is default constructible]	False
[content property]	Items
[name property]	Name
[xml lang property]	Language
(properties)	
IsSynchronizedWithCurrentItem (4)	x:Nullable(x:Boolean)
(description)	A value that indicates whether the Selector should keep the SelectedItem synchronized with the current item in the Items property.
[text syntax]	NullableBoolSyntax
SelectedIndex	x:Int32
(description)	The index of the selected item.
SelectedItem	x:Object
(description)	The selected item.

ComboBox ListBox	
SelectedValue (4)	x:Object
(description)	The value of the selected item, obtained by using the SelectedValuePath.
SelectedValuePath (4)	x:String
(description)	The property path that is used to get the SelectedValue property of the SelectedItem property.
(events)	
SelectionChanged	Occurs when the currently selected item changes.

Setter

x:Object > DependencyObject > SetterBase > Setter

(usage)	<Setter />
(description)	Applies a value to a property in a Style.
(properties)	
Value	x:Object
(description)	The value to apply to the property that is specified by the Setter.

SetterBase

x:Object > DependencyObject > SetterBase

Setter	
(usage)	None.
(description)	Represents the base type for value setters.
(used by)	SetterBaseCollection
[is default constructible]	False

SetterBaseCollection

x:Object > DependencyObject > PresentationFrameworkCollection(SetterBase) > SetterBaseCollection

(usage)	<SetterBaseCollection>**SetterBase***</SetterBaseCollection>
(description)	Represents a collection of objects that inherit from SetterBase.
(used by)	Style
[is list]	True

(usage)	<SetterBaseCollection>**SetterBase***</SetterBaseCollection>
[allowed types]	SetterBase

Shape

x:Object > DependencyObject > UIElement > FrameworkElement > Shape

Ellipse Line Path Polygon Polyline Rectangle	
(usage)	None.
(description)	Provides a base type for shape elements, such as Ellipse, Polygon, and Rectangle.
[is default constructible]	False
[name property]	Name
[xml lang property]	Language
(properties)	
Fill	Brush
(description)	The Brush that specifies how to paint the interior of the shape.
Stretch	Stretch
(description)	A Stretch enumeration value that describes how the shape fills its allocated space.
Stroke	Brush
(description)	The Brush that specifies how the Shape outline is painted.
StrokeDashArray	DoubleCollection
(description)	A collection of Double values that indicate the pattern of dashes and gaps that is used to outline shapes.
StrokeDashCap	PenLineCap
(description)	A PenLineCap enumeration value that specifies how the ends of a dash are drawn.
StrokeDashOffset	x:Double
(description)	A Double that specifies the distance within the dash pattern where a dash begins.
StrokeEndLineCap	PenLineCap
(description)	A PenLineCap enumeration value that describes the Shape at the end of a line.
StrokeLineJoin	PenLineJoin

Ellipse Line Path Polygon Polyline Rectangle	
(description)	A PenLineJoin enumeration value that specifies the type of join that is used at the vertices of a Shape.
StrokeMiterLimit	x:Double
(description)	A limit on the ratio of the miter length to half the StrokeThickness of a Shape element.
StrokeStartLineCap	PenLineCap
(description)	A PenLineCap enumeration value that describes the Shape at the start of a Stroke.
StrokeThickness	x:Double
(description)	The width of the Shape stroke outline.

SineEase

x:Object > DependencyObject > EasingFunctionBase > SineEase, IEasingFunction

(usage)	<SineEase />
(description)	Represents an easing function that creates an animation that accelerates and/or decelerates using a sine formula (see remarks below).

Size

x:Object > Size

(usage)	<Size> string </Size>
(description)	Describes the width and height of an object.
(used by)	ArcSegment
[is nullable]	False
[text syntax]	SizeSyntax
(properties)	
Height	x:Double
(description)	The height of this instance of Size.
Width	x:Double
(description)	The width of this instance of Size.
(static properties)	
Empty	Size
(description)	A value that represents a static empty Size.

SkewTransform

x:Object > DependencyObject > GeneralTransform > Transform > SkewTransform

(usage)	<SkewTransform />
(description)	Represents a two-dimensional skew.
(properties)	
AngleX	x:Double
(description)	The x-axis skew angle, which is measured in degrees counterclockwise from the y-axis.
AngleY	x:Double
(description)	The y-axis skew angle, which is measured in degrees counterclockwise from the x-axis.
CenterX	x:Double
(description)	The x-coordinate of the transform center.
CenterY	x:Double
(description)	The y-coordinate of the transform center.

Slider

x:Object > DependencyObject > UIElement > FrameworkElement > Control > RangeBase > Slider

(usage)	<Slider />
(description)	Represents a control that lets the user select from a range of values by moving a Thumb control along a track.
[name property]	Name
[xml lang property]	Language
(properties)	
IsDirectionReversed	x:Boolean
(description)	A value that indicates the direction of increasing value.
IsFocused	x:Boolean
(description)	A value indicating whether the slider control has focus.
Orientation	Orientation
(description)	The orientation of a Slider.

SolidColorBrush

x:Object > DependencyObject > Brush > SolidColorBrush

(usage)	<SolidColorBrush> string |**Color**</SolidColorBrush>
(description)	Paints an area with a solid color.
[text syntax]	SolidColorBrushSyntax
[content property]	Color
(properties)	
Color	Color
(description)	The color of this SolidColorBrush.

Span (4)

x:Object > DependencyObject > TextElement > Inline > Span

Bold Hyperlink Italic Underline	
(usage)	<Span>[Inline|x:String|UIElement]*</Span>
(description)	Groups other Inline content elements.
[content property]	Inlines
[xml lang property]	Language
(properties)	
Inlines	InlineCollection
(description)	An InlineCollection containing the top-level inline elements that include the contents of Span.
[read only]	True

SplineColorKeyFrame

x:Object > DependencyObject > ColorKeyFrame > SplineColorKeyFrame

(usage)	<SplineColorKeyFrame />
(description)	Animates from the Color value of the previous key frame to its own Value using splined interpolation.
(properties)	
KeySpline	KeySpline
(description)	The two control points that define animation progress for this key frame.

SplineDoubleKeyFrame

x:Object > DependencyObject > DoubleKeyFrame > SplineDoubleKeyFrame

(usage)	<SplineDoubleKeyFrame />
(description)	Animates from the Double value of the previous key frame to its own Value using splined interpolation.
(properties)	
KeySpline	KeySpline
(description)	The two control points that define animation progress for this key frame.

SplinePointKeyFrame

x:Object > DependencyObject > PointKeyFrame > SplinePointKeyFrame

(usage)	<SplinePointKeyFrame />
(description)	Animates from the Point value of the previous key frame to its own Value using splined interpolation.
(properties)	
KeySpline	KeySpline
(description)	The two control points that define animation progress for this key frame.

StackPanel

x:Object > DependencyObject > UIElement > FrameworkElement > Panel > StackPanel

(usage)	<StackPanel>**UIElement***</StackPanel>
(description)	Arranges child elements into a single line that can be oriented horizontally or vertically.
[content property]	Children
[name property]	Name
[xml lang property]	Language
(properties)	
Orientation	Orientation
(description)	The dimension by which child elements are stacked.

StaticResourceExtension (4)

x:Object > x:MarkupExtension > StaticResourceExtension

(usage)	{StaticResource } | <StaticResource />
(description)	Implements a markup extension that supports static (XAML load time) resource references made from XAML.

(usage)	{StaticResource } | <StaticResource />
[return value type]	x:Object
[constructors]	
(1 parameter)	
resourceKey	x:Object
(description)	

Storyboard

x:Object > DependencyObject > Timeline > Storyboard

(usage)	<Storyboard>**Timeline***</Storyboard>
(description)	Controls animations with a timeline, and provides object and property targeting information for its child animations.
(used by)	BeginStoryboard VisualState VisualTransition
[content property]	Children
(properties)	
Children	TimelineCollection
(description)	The collection of child Timeline objects.
[read only]	True
(attachable properties)	
Storyboard.Target	DependencyObject
(description)	
[target type]	Timeline
Storyboard.TargetName	x:String
(description)	The name of the object to animate.
[target type]	Timeline
Storyboard.TargetProperty	PropertyPath
(description)	The name of the property that should be animated.
[target type]	Timeline

Stretch

x:Object > Stretch

(usage)	**None**| **Fill**| **Uniform**| **UniformToFill**
(description)	Describes how content is resized to fill its allocated space.
(used by)	Image MediaElement Shape TileBrush Viewbox
[is nullable]	False
[text syntax]	StretchSyntax

StretchDirection (4)

x:Object > StretchDirection

(usage)	**UpOnly**| **DownOnly**| **Both**
(description)	Describes the direction that content is scaled.
(used by)	Viewbox
[is nullable]	False
[text syntax]	StretchDirectionSyntax

Stroke

x:Object > DependencyObject > Stroke

(usage)	<Stroke />
(description)	Represents a collection of points that correspond to a stylus-down, move, and stylus-up sequence.
(used by)	StrokeCollection
(properties)	
DrawingAttributes	DrawingAttributes
(description)	The properties of the stroke, such as Height, Width, Color, or OutlineColor.
StylusPoints	StylusPointCollection
(description)	The stylus points of the Stroke.

StrokeCollection

x:Object > DependencyObject > PresentationFrameworkCollection(Stroke) > StrokeCollection

(usage)	<StrokeCollection>**Stroke***</StrokeCollection>
(description)	Represents a collection of Stroke objects.
(used by)	InkPresenter
[is list]	True

(usage)	<StrokeCollection>**Stroke***</StrokeCollection>
[allowed types]	Stroke

Style

x:Object > DependencyObject > Style

(usage)	<Style>**SetterBase***</Style>
(description)	Contains property setters that can be shared between instances of a type.
(used by)	ComboBox FrameworkElement ListBox Style
[content property]	Setters
[dictionary key property]	TargetType
(properties)	
BasedOn	Style
(description)	A defined style that is the basis of the current style.
Setters	SetterBaseCollection
(description)	A collection of Setter objects.
[read only]	True
TargetType	x:XamlType
(description)	The type for which the style is intended.

StyleSimulations

x:Object > StyleSimulations

(usage)	**None**| **BoldSimulation**| **ItalicSimulation**| **BoldItalicSimulation**
(description)	Describes the simulation style of a font.
(used by)	Glyphs
[is nullable]	False
[text syntax]	StyleSimulationsSyntax

StylusPoint

x:Object > StylusPoint

(usage)	<StylusPoint />
(description)	Represents a single point collected while the user is entering ink strokes with the stylus or mouse.

(usage)	<StylusPoint />
(used by)	StylusPointCollection
[is nullable]	False
(properties)	
PressureFactor	x:Single
(description)	The pressure factor of the stylus on the screen.
X	x:Double
(description)	The value for the x-coordinate of the StylusPoint.
Y	x:Double
(description)	The value for the y-coordinate of the StylusPoint.

StylusPointCollection

x:Object > DependencyObject > PresentationFrameworkCollection(StylusPoint) > StylusPointCollection

(usage)	<StylusPointCollection>**StylusPoint***</StylusPointCollection>
(description)	Represents a collection of related StylusPoint objects.
(used by)	Stroke
[is list]	True
[allowed types]	StylusPoint

SupportedTextSelection (4)

x:Object > SupportedTextSelection

(usage)	**None**| **Single**| **Multiple**
(description)	Contains values that specify whether a text provider supports selection and, if so, whether it supports a single, continuous selection or multiple, disjoint selections.
[is nullable]	False
[text syntax]	SupportedTextSelectionSyntax

SweepDirection

x:Object > SweepDirection

(usage)	**Counterclockwise**| **Clockwise**
(description)	Specifies the direction an elliptical arc is drawn.

(usage)	**Counterclockwise**| **Clockwise**
(used by)	ArcSegment
[is nullable]	False
[text syntax]	SweepDirectionSyntax

SystemColors

x:Object > SystemColors

(usage)	{x:Static SystemColors.StaticPropertyName}
(description)	Contains system colors, system brushes, and system resource keys that correspond to system display elements.
[is default constructible]	False
(static properties)	
ActiveBorderColor	Color
(description)	A Color structure that is the color of the active window's border.
ActiveCaptionColor	Color
(description)	A Color structure that is the background color of the active window's title bar.
ActiveCaptionTextColor	Color
(description)	A Color structure that is the color of the text in the active window's title bar.
AppWorkspaceColor	Color
(description)	A Color structure that is the color of the application workspace.
ControlColor	Color
(description)	A Color structure that is the face color of a three-dimensional display element.
ControlDarkColor	Color
(description)	A Color structure that is the shadow color of a three-dimensional display element.
ControlDarkDarkColor	Color
(description)	A Color structure that is the dark shadow color of a three-dimensional display element.
ControlLightColor	Color
(description)	A Color structure that is the light color of a three-dimensional display element.

(usage)	{x:Static SystemColors.StaticPropertyName}
ControlLightLightColor	Color
(description)	A Color structure that is the highlight color of a three-dimensional display element.
ControlTextColor	Color
(description)	A Color structure that is the color of text in a three-dimensional display element.
DesktopColor	Color
(description)	A Color structure that is the color of the desktop.
GrayTextColor	Color
(description)	A Color structure that is the color of disabled text.
HighlightColor	Color
(description)	A Color structure that is the background color of selected items.
HighlightTextColor	Color
(description)	A Color structure that is the color of the text of selected items.
InactiveBorderColor	Color
(description)	A Color structure that is the color of an inactive window's border.
InactiveCaptionColor	Color
(description)	A Color structure that is the background color of an inactive window's title bar.
InactiveCaptionTextColor	Color
(description)	A Color structure that is the color of the text of an inactive window's title bar.
InfoColor	Color
(description)	A Color structure that is the background color for the ToolTip control.
InfoTextColor	Color
(description)	A Color structure that is the text color for the ToolTip control.
MenuColor	Color
(description)	A Color structure that is the color of a menu's background.
MenuTextColor	Color
(description)	A Color structure that is the color of a menu's text.
ScrollBarColor	Color

(usage)	{x:Static SystemColors.StaticPropertyName}
(description)	A Color structure that is the background color of a scroll bar.
WindowColor	Color
(description)	A Color structure that is the background color in the client area of a window.
WindowFrameColor	Color
(description)	A Color structure that is the color of a window frame.
WindowTextColor	Color
(description)	A Color structure that is the color of the text in the client area of a window.

SystemParameters

x:Object > SystemParameters

(usage)	{x:Static SystemParameters.StaticPropertyName}
(description)	Contains properties that you can use to query system settings.
[is default constructible]	False
(static properties)	
HighContrast	x:Boolean
(description)	A value that indicates whether the client computer is in high-contrast mode.
WheelScrollLines (4)	x:Int32
(description)	A value that indicates the number of lines to scroll vertically in response to mouse wheel events.

TabletDeviceType

x:Object > TabletDeviceType

(usage)	**Mouse**| **Stylus**| **Touch**
(description)	Defines values for the type of devices the tablet device uses.
[is nullable]	False
[text syntax]	TabletDeviceTypeSyntax

TemplateBindingExtension (4)

x:Object > x:MarkupExtension > TemplateBindingExtension

(usage)	{TemplateBinding } | <TemplateBinding />

(usage)	{TemplateBinding } | <TemplateBinding />
(description)	Implements a markup extension that supports the binding between the value of a property in a template and the value of some other exposed property on the templated control.
[return value type]	x:Object
[constructors]	
(1 parameter)	
property	DependencyProperty
(description)	

TextAlignment

x:Object > TextAlignment

(usage)	**Center**| **Left**| **Right**| **Justify**
(description)	Specifies whether text is centered, left-aligned, or right-aligned.
(used by)	Block RichTextBox TextBlock TextBox
[is nullable]	False
[text syntax]	TextAlignmentSyntax

TextBlock

x:Object > DependencyObject > UIElement > FrameworkElement > TextBlock

(usage)	<TextBlock>[**Inline**|**x:String**|**UIElement**]*</TextBlock>
(description)	Provides a lightweight control for displaying small amounts of text..
[content property]	Inlines
[name property]	Name
[xml lang property]	Language
(properties)	
FontFamily	FontFamily
(description)	The preferred top-level font family for the text content in this element.
FontSize	x:Double
(description)	The font size for the text content in this element.
FontStretch	FontStretch

(usage)	<TextBlock>[**Inline**|**x:String**|**UIElement**]*</TextBlock>
(description)	The font stretch for the text content in this element.
FontStyle	FontStyle
(description)	The font style for the content in this element.
FontWeight	FontWeight
(description)	The top-level font weight for the TextBlock.
Foreground	Brush
(description)	The Brush to apply to the text contents of the TextBlock.
Inlines	InlineCollection
(description)	The collection of inline text elements within a TextBlock.
[read only]	True
LineHeight	x:Double
(description)	The height of each line of content.
LineStackingStrategy	LineStackingStrategy
(description)	A value that indicates how a line box is determined for each line of text in the TextBlock.
Padding	Thickness
(description)	A value that indicates the thickness of padding space between the boundaries of the content area and the content displayed by a TextBlock.
Text	x:String
(description)	The text contents of a TextBlock.
TextAlignment	TextAlignment
(description)	A value that indicates the horizontal alignment of text content.
TextDecorations	TextDecorationCollection
(description)	A value that specifies the text decorations that are applied to the content in a TextBlock element.
TextTrimming (4)	TextTrimming
(description)	The text trimming behavior to employ when content overflows the content area.
TextWrapping	TextWrapping
(description)	How the TextBlock wraps text.

TextBox

x:Object > DependencyObject > UIElement > FrameworkElement > Control > TextBox

(usage)	<TextBox />
(description)	Represents a control that can be used to display single-format, multi-line text.
[name property]	Name
[xml lang property]	Language
(properties)	
AcceptsReturn	x:Boolean
(description)	The value that determines whether the text box allows and displays the newline or return characters.
CaretBrush	Brush
(description)	The brush that is used to render the vertical bar that indicates the insertion point.
HorizontalScrollBarVisibility	ScrollBarVisibility
(description)	The visibility of the horizontal scroll bar.
InputScope (4)	InputScope
(description)	The context for input use by this TextBox.
IsReadOnly	x:Boolean
(description)	The value that determines if the user can change the text in the text box.
MaxLength	x:Int32
(description)	The value that determines the maximum number of characters allowed for user input.
SelectedText	x:String
(description)	The content of the current selection in the text box.
SelectionBackground	Brush
(description)	The brush that fills the background of the selected text.
SelectionForeground	Brush
(description)	The brush used for the selected text in the text box.
SelectionLength	x:Int32
(description)	The number of characters in the current selection in the text box.
SelectionStart	x:Int32
(description)	The starting position of the text selected in the text box.
Text	x:String

(usage)	<TextBox />
(description)	The text contents of the text box.
TextAlignment	TextAlignment
(description)	How the text should be aligned in the text box.
TextWrapping	TextWrapping
(description)	How line breaking occurs if a line of text extends beyond the available width of the text box.
VerticalScrollBarVisibility	ScrollBarVisibility
(description)	The visibility of the vertical scroll bar.
Watermark (4)	x:Object
(description)	The content displayed as a watermark in the TextBox when it is empty.
(events)	
SelectionChanged	Occurs when the text selection has changed.
TextChanged	Occurs when content changes in the text box.

TextDecorationCollection

x:Object > TextDecorationCollection

(usage)	<TextDecorationCollection> string </TextDecorationCollection>
(description)	Provides the value for the TextDecorations and TextDecorations properties.
(used by)	Hyperlink Inline TextBlock
[is default constructible]	False
[text syntax]	TextDecorationCollectionSyntax

TextDecorations

x:Object > TextDecorations

(usage)	{x:Static TextDecorations.StaticPropertyName}
(description)	Implements a set of predefined text decorations.
[is default constructible]	False
(static properties)	
Underline	TextDecorationCollection
(description)	Specifies an underlined text decoration.

TextElement (4)

x:Object > DependencyObject > TextElement

Block Inline	
(usage)	None.
(description)	An base type for the abstract Block and Inline classes.
[is default constructible]	False
(properties)	
FontFamily	FontFamily
(description)	The preferred top-level font family for the content of the element.
FontSize	x:Double
(description)	The font size for the content of the element.
FontStretch	FontStretch
(description)	The glyph width of the font in a family to select.
FontStyle	FontStyle
(description)	The font style for the content in this element.
FontWeight	FontWeight
(description)	The top-level font weight to select from the font family for the content in this element.
Foreground	Brush
(description)	The Brush to apply to the content in this element.

TextElementCollection(T) (4)

x:Object > DependencyObject > PresentationFrameworkCollection(TextElementCollection+T) > TextElementCollection(T)

(usage)	None.
(description)	Provides standard facilities for creating and managing a type-safe, ordered collection of TextElement objects.
[is default constructible]	False
[is list]	True
[allowed types]	T

TextHintingMode

x:Object > TextHintingMode

(usage)	**Fixed**| **Animated**
(description)	Specifies whether text rendering is optimized for readability or not.
(used by)	TextOptions
[is nullable]	False
[text syntax]	TextHintingModeSyntax

TextOptions

x:Object > TextOptions

(usage)	None.
(description)	Provides options for controlling the rendering behavior of text.
[is default constructible]	False
(attachable properties)	
TextOptions.TextHintingMode	TextHintingMode
(description)	A value that indicates whether text rendering is optimized for readability.
[target type]	FrameworkElement

TextTrimming (4)

x:Object > TextTrimming

(usage)	**None**| **WordEllipsis**
(description)	Describes how text is trimmed when it overflows the edge of its containing box.
(used by)	TextBlock
[is nullable]	False
[text syntax]	TextTrimmingSyntax

TextWrapping

x:Object > TextWrapping

(usage)	**Wrap**| **NoWrap**
(description)	Specifies whether text wraps when it reaches the edge of its container.
(used by)	RichTextBox TextBlock TextBox

(usage)	**Wrap| NoWrap**
[is nullable]	False
[text syntax]	TextWrappingSyntax

Thickness

x:Object > Thickness

(usage)	<Thickness> string </Thickness>
(description)	Describes the thickness of a frame around a rectangle. Four Double values describe the Left, Top, Right, and Bottom sides of the rectangle, respectively.
(used by)	Border Control FrameworkElement TextBlock
[is nullable]	False
[text syntax]	ThicknessSyntax
(properties)	
Bottom	x:Double
(description)	The width, in pixels, of the lower side of the bounding rectangle.
Left	x:Double
(description)	The width, in pixels, of the left side of the bounding rectangle.
Right	x:Double
(description)	The width, in pixels, of the right side of the bounding rectangle.
Top	x:Double
(description)	The width, in pixels, of the upper side of the bounding rectangle.

Thumb

x:Object > DependencyObject > UIElement > FrameworkElement > Control > Thumb

(usage)	<Thumb />
(description)	Represents a control that can be dragged by the user.
[name property]	Name
[xml lang property]	Language
(properties)	
IsDragging	x:Boolean

(usage)	<Thumb />
(description)	Whether the Thumb control has focus and mouse capture.
IsFocused	[x:Boolean]()
(description)	Whether the thumb has focus.
(events)	
DragCompleted	Occurs when the Thumb control loses mouse capture.
DragDelta	Occurs one or more times as the mouse pointer is moved when a Thumb control has logical focus and mouse capture.
DragStarted	Occurs when a Thumb control receives logical focus and mouse capture.

TileBrush

[x:Object]() > [DependencyObject]() > [Brush]() > TileBrush

[ImageBrush]() [VideoBrush]() [WebBrowserBrush]()	
(usage)	None.
(description)	Base type that describes a way to paint a region.
[is default constructible]	False
(properties)	
AlignmentX	[AlignmentX]()
(description)	The horizontal alignment of content in the TileBrush base tile.
AlignmentY	[AlignmentY]()
(description)	The vertical alignment of content in the TileBrush base tile.
Stretch	[Stretch]()
(description)	A value that specifies how the content of this TileBrush stretches to fit its tiles.

Timeline

[x:Object]() > [DependencyObject]() > Timeline

[ColorAnimation]() [ColorAnimationUsingKeyFrames]() [DoubleAnimation]() [DoubleAnimationUsingKeyFrames]() [ObjectAnimationUsingKeyFrames]() [PointAnimation]() [PointAnimationUsingKeyFrames]() [Storyboard]()	
(usage)	None.
(description)	Defines a segment of time.

ColorAnimation ColorAnimationUsingKeyFrames DoubleAnimation DoubleAnimationUsingKeyFrames ObjectAnimationUsingKeyFrames PointAnimation PointAnimationUsingKeyFrames Storyboard	
(used by)	TimelineCollection
[is default constructible]	False
(properties)	
AutoReverse	x:Boolean
(description)	A value that indicates whether the timeline plays in reverse after it completes a forward iteration.
BeginTime	x:Nullable(x:TimeSpan)
(description)	The time at which this Timeline should begin.
Duration	Duration
(description)	The length of time for which this timeline plays, not counting repetitions.
FillBehavior	FillBehavior
(description)	A value that specifies how the animation behaves after it reaches the end of its active period.
RepeatBehavior	RepeatBehavior
(description)	The repeating behavior of this timeline.
SpeedRatio	x:Double
(description)	The rate, relative to its parent, at which time progresses for this Timeline.
(events)	
Completed	Occurs when the Storyboard object has completed playing.

TimelineCollection

x:Object > DependencyObject > PresentationFrameworkCollection(Timeline) > TimelineCollection

(usage)	<TimelineCollection>**Timeline***</TimelineCollection>
(description)	Represents a collection of Timeline objects.
(used by)	Storyboard
[is list]	True
[allowed types]	Timeline

TimelineMarker

x:Object > DependencyObject > TimelineMarker

(usage)	<TimelineMarker />
(description)	Represents metadata associated with a specific point in a media file.
(used by)	TimelineMarkerCollection
(properties)	
Text	x:String
(description)	The text value of a TimelineMarker.
Time	x:TimeSpan
(description)	The time at which a TimelineMarker is reached.
Type	x:String
(description)	The marker type of a TimelineMarker.

TimelineMarkerCollection

x:Object > DependencyObject > PresentationFrameworkCollection(TimelineMarker) > TimelineMarkerCollection

(usage)	<TimelineMarkerCollection>**TimelineMarker***</TimelineMarkerCollection>
(description)	Represents a collection of TimelineMarker objects that can be individually accessed by index.
(used by)	MediaElement
[is list]	True
[allowed types]	TimelineMarker

ToggleButton

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ButtonBase > ToggleButton

CheckBox RadioButton	
(usage)	<ToggleButton>x:Object</ToggleButton>
(description)	Base type for controls that can switch states, such as CheckBox and RadioButton.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
IsChecked	x:Nullable(x:Boolean)

CheckBox RadioButton	
(description)	Whether the ToggleButton is checked.
[text syntax]	NullableBoolSyntax
IsThreeState	x:Boolean
(description)	Whether the control supports two or three states.
(events)	
Checked	Occurs when a ToggleButton is checked.
Indeterminate	Occurs when the state of a ToggleButton is switched to the indeterminate state.
Unchecked	Occurs when a ToggleButton is unchecked.

ToggleState

x:Object > ToggleState

(usage)	**Off**| **On**| **Indeterminate**
(description)	Contains values that specify the ToggleState of a UI automation element
[is nullable]	False
[text syntax]	ToggleStateSyntax

ToolTip

x:Object > DependencyObject > UIElement > FrameworkElement > Control > ContentControl > ToolTip

(usage)	<ToolTip>**x:Object**</ToolTip>
(description)	Represents a control that creates a pop-up window that displays information for an element in the UI.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
HorizontalOffset	x:Double
(description)	The horizontal distance between the target origin and the pop-up alignment point.
IsOpen	x:Boolean
(description)	A value that indicates whether the ToolTip is visible.

(usage)	<ToolTip>**x:Object**</ToolTip>
Placement	PlacementMode
(description)	How the ToolTip should be positioned in relation to the PlacementTarget.
PlacementTarget	UIElement
(description)	The visual element or control that the tool tip should be positioned in relation to when opened by the ToolTipService.
VerticalOffset	x:Double
(description)	The vertical distance between the target origin and the pop-up alignment point.
(events)	
Closed	Occurs when a ToolTip is closed and is no longer visible.
Opened	Occurs when a ToolTip becomes visible.

ToolTipService

x:Object > ToolTipService

(usage)	None.
(description)	Represents a service that provides static methods to display a tooltip.
[is default constructible]	False
(attachable properties)	
ToolTipService.Placement	PlacementMode
(description)	The position of the tooltip in relation to its target visual element.
[target type]	DependencyObject
ToolTipService.PlacementTarget	UIElement
(description)	The visual element that the tooltip should be positioned in relation to.
[target type]	DependencyObject
ToolTipService.ToolTip	x:Object
(description)	A tooltip to be attached to a control.
[target type]	DependencyObject

TouchAction

x:Object > TouchAction

(usage)	**Down**| **Move**| **Up**
(description)	Describes the action of a specific touch point.
[is nullable]	False
[text syntax]	TouchActionSyntax

TouchDevice

x:Object > DependencyObject > TouchDevice

(usage)	<TouchDevice>**x:Object**</TouchDevice>
(description)	Identifies a "device" that produced the touch point. This is not a literal device, instead it distinguishes touch points through a down/move/up sequence.
[content property]	DirectlyOver

TouchPoint

x:Object > DependencyObject > TouchPoint

(usage)	<TouchPoint />
(description)	Represents a single touch point from a multitouch message source.
(used by)	TouchPointCollection

TouchPointCollection

x:Object > DependencyObject > PresentationFrameworkCollection(TouchPoint) > TouchPointCollection

(usage)	None.
(description)	Contains a collection of TouchPoint values.
[is default constructible]	False
[is list]	True
[allowed types]	TouchPoint

Transform

x:Object > DependencyObject > GeneralTransform > Transform

CompositeTransform MatrixTransform RotateTransform ScaleTransform SkewTransform TransformGroup TranslateTransform	
(usage)	<Transform> string </Transform>

CompositeTransform MatrixTransform RotateTransform ScaleTransform SkewTransform TransformGroup TranslateTransform	
(description)	Defines functionality that enables transformations in a two-dimensional plane.
(used by)	Brush Geometry TransformCollection UIElement
[is default constructible]	False
[text syntax]	TransformSyntax

TransformCollection

x:Object > DependencyObject > PresentationFrameworkCollection(Transform) > TransformCollection

(usage)	<TransformCollection>**Transform***</TransformCollection>
(description)	Represents a collection of Transform objects that can be individually accessed by index.
(used by)	TransformGroup
[is list]	True
[allowed types]	Transform

TransformGroup

x:Object > DependencyObject > GeneralTransform > Transform > TransformGroup

(usage)	<TransformGroup> string |**Transform***</TransformGroup>
(description)	Represents a composite Transform composed of other Transform objects.
[text syntax]	TransformGroupSyntax
[content property]	Children
(properties)	
Children	TransformCollection
(description)	The collection of child Transform objects.

TranslateTransform

x:Object > DependencyObject > GeneralTransform > Transform > TranslateTransform

(usage)	<TranslateTransform />
(description)	Translates (moves) an object in the two-dimensional x-y coordinate system.
(properties)	

(usage)	<TranslateTransform />
X	x:Double
(description)	The distance to translate along the x-axis.
Y	x:Double
(description)	The distance to translate (move) an object along the y-axis.

TriggerAction

x:Object > DependencyObject > TriggerAction

BeginStoryboard	
(usage)	None.
(description)	Serves as the base type for BeginStoryboard.
(used by)	TriggerActionCollection
[is default constructible]	False

TriggerActionCollection

x:Object > DependencyObject > PresentationFrameworkCollection(TriggerAction) > TriggerActionCollection

(usage)	<TriggerActionCollection>**TriggerAction***</TriggerActionCollection>
(description)	Represents a collection of BeginStoryboard objects.
(used by)	EventTrigger
[is list]	True
[allowed types]	TriggerAction

TriggerBase

x:Object > DependencyObject > TriggerBase

EventTrigger	
(usage)	None.
(description)	Serves as the base type for EventTrigger.
(used by)	TriggerCollection
[is default constructible]	False

TriggerCollection

x:Object > DependencyObject > PresentationFrameworkCollection(TriggerBase) > TriggerCollection

(usage)	None.
(description)	Represents a collection of EventTrigger objects.
(used by)	FrameworkElement
[is default constructible]	False
[is list]	True
[allowed types]	TriggerBase

UIElement

x:Object > DependencyObject > UIElement

FrameworkElement	
(usage)	None.
(description)	UIElement is a base type for most of the objects that have visual appearance and can process basic input.
(used by)	Application AutomationProperties Border InlineCollection InlineUIContainer Popup ToolTip ToolTipService UIElementCollection UserControl Viewbox
[is default constructible]	False
(properties)	
AllowDrop (4)	x:Boolean
(description)	Gets or sets a value that determines whether this UIElement can be a drop target for purposes of drag-and-drop operations.
CacheMode	CacheMode
(description)	A value that indicates that rendered content should be cached when possible.
Clip	Geometry
(description)	The Geometry used to define the outline of the contents of a UIElement.
Effect	Effect
(description)	The pixel-shader effect to use for rendering this UIElement.
IsHitTestVisible	x:Boolean
(description)	Whether the contained area of this UIElement can return true values for hit testing.
Opacity	x:Double

FrameworkElement	
(description)	The degree of the object's opacity.
OpacityMask	Brush
(description)	The brush used to alter the opacity of regions of this object.
Projection	Projection
(description)	The perspective projection (3-D effect) to apply when rendering this UIElement.
RenderTransform	Transform
(description)	Transform information that affects the rendering position of a UIElement.
RenderTransformOrigin	Point
(description)	The origin point of any possible render transform declared by RenderTransform, relative to the bounds of the UIElement.
UseLayoutRounding	x:Boolean
(description)	A value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Visibility	Visibility
(description)	The visibility of a UIElement. A UIElement that is not visible does not render and does not communicate its desired size to layout.
(events)	
DragEnter (4)	Occurs when the input system reports an underlying drag event with this element as the target.
DragLeave (4)	Occurs when the input system reports an underlying drag event with this element as the origin.
DragOver (4)	Occurs when the input system reports an underlying drag event with this element as the potential drop target.
Drop (4)	Occurs when the input system reports an underlying drop event with this element as the drop target.
GotFocus	Occurs when a UIElement receives focus.
KeyDown	Occurs when a keyboard key is pressed while the UIElement has focus.
KeyUp	Occurs when a keyboard key is released while the UIElement has focus.
LostFocus	Occurs when a UIElement loses focus.
LostMouseCapture	Occurs when the UIElement loses mouse capture.
ManipulationCompleted (4)	Occurs when a manipulation and inertia on the UIElement is complete.
ManipulationDelta (4)	Occurs when the input device changes position during a manipulation.

FrameworkElement	
ManipulationStarted (4)	Occurs when an input device begins a manipulation on the UIElement.
MouseEnter	Occurs when the mouse (or a stylus) enters the bounding area of a UIElement.
MouseLeave	Occurs when the mouse (or the stylus) leaves the bounding area of a UIElement.
MouseLeftButtonDown	Occurs when the left mouse button is pressed (or when the tip of the stylus touches the tablet) while the mouse pointer is over a UIElement.
MouseLeftButtonUp	Occurs when the left mouse button is released (or the tip of the stylus is removed from the tablet) while the mouse (or the stylus) is over a UIElement (or while a UIElement holds mouse capture).
MouseMove	Occurs when the coordinate position of the mouse (or stylus) changes while over a UIElement (or while a UIElement holds mouse capture).
MouseRightButtonDown (4)	Occurs when the right mouse button is pressed while the mouse pointer is over a UIElement.
MouseRightButtonUp (4)	Occurs when the right mouse button is released while the mouse pointer is over a UIElement. However, this event will only be raised if a caller marks the preceding MouseRightButtonDown event as handled; see Remarks.
MouseWheel	Occurs when the user rotates the mouse wheel while the mouse pointer is over a UIElement, or the UIElement has focus.
TextInput (4)	Occurs when a UI element gets text in a device-independent manner.
TextInputStart (4)	Occurs when a UI element initially gets text in a device-independent manner.
TextInputUpdate (4)	Occurs when text continues to be composed via an input method editor (IME).

UIElementCollection

x:Object > DependencyObject > PresentationFrameworkCollection(UIElement) > UIElementCollection

(usage)	None.
(description)	Represents an ordered collection of UIElement objects.
(used by)	Panel
[is default constructible]	False
[is list]	True
[allowed types]	UIElement

Underline (4)

x:Object > DependencyObject > TextElement > Inline > Span > Underline

(usage)	<Underline>[**Inline**|**x:String**|**UIElement**]*</Underline>
(description)	Provides an inline-level content element that causes content to appear with an underlined text decoration.
[content property]	Inlines
[xml lang property]	Language

UpdateSourceTrigger

x:Object > UpdateSourceTrigger

(usage)	**Default**| **Explicit**
(description)	Defines constants that indicate when a binding source is updated by its binding target in two-way binding.
(used by)	Binding
[is nullable]	False
[text syntax]	UpdateSourceTriggerSyntax

UserControl

x:Object > DependencyObject > UIElement > FrameworkElement > Control > UserControl

(usage)	<UserControl>**UIElement**</UserControl>
(description)	Provides the base type for defining a new control that encapsulates related existing controls and provides its own logic.
[content property]	Content
[name property]	Name
[xml lang property]	Language
(properties)	
Content	UIElement
(description)	The content that is contained within a user control.

ValidationErrorEventAction

x:Object > ValidationErrorEventAction

(usage)	**Added**| **Removed**

(usage)	**Added**| **Removed**
(description)	Describes the reason a BindingValidationError event has occurred.
[is nullable]	False
[text syntax]	ValidationErrorEventActionSyntax

VerticalAlignment

x:Object > VerticalAlignment

(usage)	**Top**| **Center**| **Bottom**| **Stretch**
(description)	Describes how a child element is vertically positioned or stretched within a parent's layout slot.
(used by)	Control FrameworkElement
[is nullable]	False
[text syntax]	VerticalAlignmentSyntax

VideoBrush

x:Object > DependencyObject > Brush > TileBrush > VideoBrush

(usage)	<VideoBrush />
(description)	Paints an area with video content.
(properties)	
SourceName	x:String
(description)	The name of the MediaElement to use as the source of the VideoBrush.

VideoCaptureDeviceCollection (4)

x:Object > DependencyObject > PresentationFrameworkCollection(VideoCaptureDevice) > VideoCaptureDeviceCollection

(usage)	None.
(description)	Specifies a collection of VideoCaptureDevice objects.
[is default constructible]	False
[is list]	True
[allowed types]	VideoCaptureDevice

VideoOutputConnectorType (4)

x:Object > VideoOutputConnectorType

(usage)	**Other**| **Vga**| **SVideo**| **CompositeVideo**| **ComponentVideo**| **Dvi**| **Hdmi**| **Llvds**| **JapaneseDConnector**...
(description)	Defines of all of the commonly available output connector types.
[is nullable]	False
[text syntax]	VideoOutputConnectorTypeSyntax

VideoSink (4)

x:Object > VideoSink

(usage)	<VideoSink />
(description)	Exposes the capture graph for video devices. Derive from this type to receive video information and to obtain the capture graph through CaptureSource.
(properties)	
CaptureSource	CaptureSource
(description)	A capture source that is associated with this VideoSink.

Viewbox (4)

x:Object > DependencyObject > UIElement > FrameworkElement > Viewbox

(usage)	<Viewbox>**UIElement**</Viewbox>
(description)	Defines a content decorator that can stretch and scale a single child to fill the available space.
[content property]	Child
[name property]	Name
[xml lang property]	Language
(properties)	
Child	UIElement
(description)	The single child element of a Viewbox element.
Stretch	Stretch
(description)	The Stretch mode, which determines how content fits into the available space.
StretchDirection	StretchDirection

(usage)	<Viewbox>**UIElement**</Viewbox>
(description)	The StretchDirection, which determines how scaling is applied to the contents of a Viewbox.

VirtualizationMode

x:Object > VirtualizationMode

(usage)	**Standard**| **Recycling**
(description)	Specifies the method the VirtualizingStackPanel uses to manage virtualizing its child items.
(used by)	VirtualizingStackPanel
[is nullable]	False
[text syntax]	VirtualizationModeSyntax

VirtualizingPanel

x:Object > DependencyObject > UIElement > FrameworkElement > Panel > VirtualizingPanel

VirtualizingStackPanel	
(usage)	None.
(description)	Provides a framework for Panel elements that virtualize their visual children.
[is default constructible]	False
[content property]	Children
[name property]	Name
[xml lang property]	Language

VirtualizingStackPanel

x:Object > DependencyObject > UIElement > FrameworkElement > Panel > VirtualizingPanel > VirtualizingStackPanel

(usage)	<VirtualizingStackPanel>**UIElement***</VirtualizingStackPanel >
(description)	Arranges and virtualizes content on a single line that is oriented either horizontally or vertically.
[content property]	Children
[name property]	Name
[xml lang property]	Language
(properties)	

(usage)	<VirtualizingStackPanel>**UIElement***</VirtualizingStackPanel>
CanHorizontallyScroll	x:Boolean
(description)	A value that indicates whether a VirtualizingStackPanel can scroll in the horizontal dimension.
CanVerticallyScroll	x:Boolean
(description)	A value that indicates whether content can scroll in the vertical dimension.
Orientation	Orientation
(description)	A value that describes the horizontal or vertical orientation of stacked content.
ScrollOwner	ScrollViewer
(description)	A value that identifies the container that controls scrolling behavior in this VirtualizingStackPanel.
(attachable properties)	
VirtualizingStackPanel.VirtualizationMode (4)	VirtualizationMode
(description)	Specifies how a panel in an ItemsControl virtualizes its child items.
[target type]	DependencyObject
(events)	
CleanUpVirtualizedItemEvent	Occurs when an item that is hosted by the VirtualizingStackPanel is re-virtualized.

Visibility

x:Object > Visibility

(usage)	**Visible**| **Collapsed**
(description)	Specifies the display state of an element.
(used by)	ScrollViewer UIElement
[is nullable]	False
[text syntax]	VisibilitySyntax

VisualState

x:Object > DependencyObject > VisualState

(usage)	

(usage)	
(description)	Represents the visual appearance of the control when it is in a specific state.
[content property]	Storyboard
(properties)	
Storyboard	Storyboard
(description)	A Storyboard that defines the appearance of the control when it is the state that is represented by the VisualState.

VisualStateGroup

x:Object > DependencyObject > VisualStateGroup

(usage)	<VisualStateGroup>**x:Object***</VisualStateGroup>
(description)	Contains mutually exclusive VisualState objects and VisualTransition objects that are used to go from one state to another.
[content property]	States
(properties)	
States	IList
(description)	The collection of mutually exclusive VisualState objects.
[read only]	True
Transitions	IList
(description)	The collection of VisualTransition objects.
[read only]	True
(events)	
CurrentStateChanged	Occurs after a control transitions into a different state.
CurrentStateChanging	Occurs when a control begins transitioning into a different state.

VisualStateManager

x:Object > DependencyObject > VisualStateManager

(usage)	<VisualStateManager />
(description)	Manages states and the logic for transitioning between states for controls.
(used by)	VisualStateManager
(attachable properties)	

(usage)	<VisualStateManager />
VisualStateManager.CustomVisualStateManager	VisualStateManager
(description)	The VisualStateManager that transitions between the states of a control.
[target type]	FrameworkElement
VisualStateManager.VisualStateGroups	IList
(description)	
[target type]	FrameworkElement
[read only]	True

VisualTransition

x:Object > DependencyObject > VisualTransition

(usage)	<VisualTransition>**Storyboard**</VisualTransition>
(description)	Represents the visual behavior that occurs when the control transitions from one state to another.
[content property]	Storyboard
(properties)	
From	x:String
(description)	The name of the VisualState to transition from.
GeneratedDuration	Duration
(description)	The amount of time it takes to move from one state to another.
GeneratedEasingFunction	IEasingFunction
(description)	The easing function applied to the generated animations.
Storyboard	Storyboard
(description)	The Storyboard that occurs when the transition occurs.
To	x:String
(description)	The name of the VisualState to transition to.

WaveFormatType (4)

x:Object > WaveFormatType

(usage)	**Pcm**
(description)	Reports the digital signal encoding format of an audio format.

(usage)	**Pcm**
[is nullable]	False
[text syntax]	WaveFormatTypeSyntax

WebBrowser (4)

x:Object > DependencyObject > UIElement > FrameworkElement > WebBrowser

(usage)	<WebBrowser />
(description)	Hosts HTML content.
[name property]	Name
[xml lang property]	Language
(properties)	
Source	x:Uri
(description)	The URI source of the HTML content to display in the WebBrowser control.
(events)	
LoadCompleted	Occurs when top-level navigation completes and the content loads into the WebBrowser control or when an error occurs during loading.
ScriptNotify	Occurs when the content contained in the WebBrowser control passes a string by using JavaScript.

WebBrowserBrush (4)

x:Object > DependencyObject > Brush > TileBrush > WebBrowserBrush

(usage)	<WebBrowserBrush />
(description)	Provides a brush that renders the currently hosted HTML.
(properties)	
SourceName	x:String
(description)	The name of the source WebBrowser control that provides the HTML content.

Window (4)

x:Object > DependencyObject > Window

(usage)	<Window />
(description)	Represents an out-of-browser application window.
(properties)	

(usage)	<Window />
Height	x:Double
(description)	The height of the application window in pixels.
Left	x:Double
(description)	The position of the left edge of the application window; see Remarks for restrictions on setting this property at run time.
Top	x:Double
(description)	The position of the top edge of the application window; see Remarks for restrictions on setting this property at run time.
TopMost	x:Boolean
(description)	A value that indicates whether the application window is always displayed in front of other windows.
Width	x:Double
(description)	The width of the application window in pixels.
WindowState	WindowState
(description)	A value that indicates whether the window is maximized, minimized, or in the normal state.
(events)	
Closing	Occurs when the window is about to close.

WindowInteractionState

x:Object > WindowInteractionState

(usage)	**Running**| **Closing**| **ReadyForUserInteraction**| **BlockedByModalWindow**| **NotResponding**
(description)	Defines values that specify the current state of the window for purposes of user or programmatic interaction.
[is nullable]	False
[text syntax]	WindowInteractionStateSyntax

WindowResizeEdge (4)

x:Object > WindowResizeEdge

(usage)	**Left**| **Right**| **Top**| **TopLeft**| **TopRight**| **Bottom**| **BottomLeft**| **BottomRight**
(description)	Defines constants that represent the edges and corners of a out-of-browser application window.

(usage)	**Left**| **Right**| **Top**| **TopLeft**| **TopRight**| **Bottom**| **BottomLeft**| **BottomRight**
[is nullable]	False
[text syntax]	WindowResizeEdgeSyntax

WindowSettings

x:Object > DependencyObject > WindowSettings

(usage)	<WindowSettings />
(description)	Represents information about an out-of-browser application window.

WindowStartupLocation (4)

x:Object > WindowStartupLocation

(usage)	**CenterScreen**| **Manual**
(description)	Defines constants that indicate how an out-of-browser application window is positioned at startup.
[is nullable]	False
[text syntax]	WindowStartupLocationSyntax

WindowState (4)

x:Object > WindowState

(usage)	**Normal**| **Minimized**| **Maximized**
(description)	Defines constants that indicate the state of an out-of-browser application window.
(used by)	Window
[is nullable]	False
[text syntax]	WindowStateSyntax

WindowStyle (4)

x:Object > WindowStyle

(usage)	**SingleBorderWindow**| **None**| **BorderlessRoundCornersWindow**
(description)	Defines constants that indicate the appearance of the title bar and border of an out-of-browser application window.
[is nullable]	False

(usage)	**SingleBorderWindow**| **None**| **BorderlessRoundCornersWindow**
[text syntax]	WindowStyleSyntax

WindowVisualState

x:Object > WindowVisualState

(usage)	**Normal**| **Maximized**| **Minimized**
(description)	Contains values that specify the visual state of a window for the IWindowProvider pattern.
[is nullable]	False
[text syntax]	WindowVisualStateSyntax

Silverlight XamlType Information Items for Assignable Types

x:Boolean

x:Object > x:Boolean

link to externally defined type	x:Boolean, from **[MS-XAML]**

x:Byte

x:Object > x:Byte

link to externally defined type	x:Byte, from **[MS-XAML]**

x:Char

x:Object > x:Char

link to externally defined type	x:Char, from **[MS-XAML]**

Dictionary(T,U)

x:Object > Dictionary(T,U)

(usage)	<Dictionary x:TypeArguments="T">**Dictionary(x:String,x:String)**</Dictionary>
(description)	
(used by)	MediaElement
[is dictionary]	True
[allowed types]	x:String
[allowed key types]	x:String

x:Double

x:Object > x:Double

link to externally defined type	x:Double, from **[MS-XAML]**

GroupDescription

x:Object > GroupDescription

PropertyGroupDescription	
(usage)	None.
(description)	Provides a base type for defining how to divide the items in a collection into

PropertyGroupDescription	
	groups.
(used by)	CollectionViewSource
[is default constructible]	False

IEnumerable

IEnumerable

(usage)	None.
(description)	Exposes the enumerator, which supports a simple iteration over a non-generic collection.
(used by)	ItemsControl
[is default constructible]	False

IList

IList

(usage)	None.
(description)	Represents a non-generic collection of objects that can be individually accessed by index.
(used by)	Application InputScope ListBox VisualStateGroup VisualStateManager
[is default constructible]	False
[is list]	True
[allowed types]	x:Object

x:Int32

x:Object > x:Int32

link to externally defined type	x:Int32, from **[MS-XAML]**

x:MarkupExtension

x:Object > x:MarkupExtension

BindingBase RelativeSource StaticResourceExtension TemplateBindingExtension	
link to externally defined type	x:MarkupExtension, from [MS-XAML]

x:Nullable(T)

x:Object > x:Nullable(T)

link to externally defined type	x:Nullable(T), from **[MS-XAML]**

x:Object

x:Object

Analytics Application AudioSink AutomationProperties CaptureDeviceConfiguration Colors Cursor Cursors DataObject DependencyObject DomainAcquirer FontFamily Fonts FontStretches FontStyles FontWeights Keyboard LicenseAcquirer LicenseManagement OpenFileDialog PropertyPath SaveFileDialog SystemColors SystemParameters TextDecorationCollection TextDecorations TextOptions ToolTipService VideoSink	
link to externally defined type	x:Object, from [MS-XAML]

ObservableCollection(T)

x:Object > ObservableCollection(T)

(usage)	<ObservableCollection x:TypeArguments="T">T*</ObservableCollection>
(description)	
(used by)	CollectionViewSource
[is list]	True
[allowed types]	T

x:Single

x:Object > x:Single

link to externally defined type	x:Single, from **[MS-XAML]**

SortDescriptionCollection

x:Object > SortDescriptionCollection

(usage)	<SortDescriptionCollection>**SortDescription***</SortDescriptionCollection>
(description)	Represents a collection of SortDescription instances.
(used by)	CollectionViewSource
[is list]	True
[allowed types]	SortDescription

x:String

x:Object > x:String

link to externally defined type	x:String, from **[MS-XAML]**

StringComparison

x:Object > StringComparison

(usage)	**CurrentCulture**| **CurrentCultureIgnoreCase**| **InvariantCulture**| **InvariantCultureIgnoreCase**| **Ordinal**| **OrdinalIgnoreCase**
(description)	Specifies the culture, case, and sort rules to be used by certain overloads of the String) and Object) methods.
(used by)	PropertyGroupDescription
[is nullable]	False
[text syntax]	StringComparisonSyntax

x:TimeSpan

x:Object > x:TimeSpan

link to externally defined type	x:TimeSpan, from **[MS-XAML]**

x:XamlType

x:Object > x:XamlType

link to externally defined type	x:XamlType, from **[MS-XAML]**

x:Uri

x:Object > x:Uri

link to externally defined type	x:Uri, from **[MS-XAML]**

Silverlight Xaml Text Syntax Information Sets

AlignmentXSyntax

[values]	
Center	The contents align toward the center of the container.
Left	The contents align toward the left of the container.
Right	The contents align toward the right of the container.

AlignmentYSyntax

[values]	
Bottom	The contents align toward the lower edge of the container.
Center	The contents align toward the center of the container.
Top	The contents align toward the upper edge of the container.

BindingModeSyntax

[values]	
OneTime	Updates the target property when the binding is created.
OneWay	Updates the target property when the binding is created. Changes to the source object can also propagate to the target.
TwoWay	Updates either the target or the source object when either changes. When the binding is created, the target property is updated from the source.

BitmapCreateOptionsSyntax

[values]	
DelayCreation	Causes a BitmapSource object to delay initialization until it is necessary. This is useful when dealing with collections of images. This is the default value of the CreateOptions property in Silverlight.
IgnoreImageCache	Loads images without using an existing image cache. This option should only be selected when images in a cache need to be refreshed.
None	No options are specified. This is the NOT the default value for the CreateOptions property in Silverlight (DelayCreation is the default).

BrushMappingModeSyntax

[values]	
Absolute	The coordinate system is not relative to a bounding box. Values are interpreted directly in local space.

[values]	
RelativeToBoundingBox	The coordinate system is relative to a bounding box: 0 indicates 0 percent of the bounding box, and 1 indicates 100 percent of the bounding box. For example, (0.5, 0.5) describes a point in the middle of the bounding box, and (1, 1) describes a point at the bottom right of the bounding box.

BrushSyntax

[values]	
AliceBlue	The solid fill color that has a hexadecimal value of #FFF0F8FF.
AntiqueWhite	The solid fill color that has a hexadecimal value of #FFFAEBD7.
Aqua	The solid fill color that has a hexadecimal value of #FF00FFFF.
Aquamarine	The solid fill color that has a hexadecimal value of #FF7FFFD4.
Azure	The solid fill color that has a hexadecimal value of #FFF0FFFF.
Beige	The solid fill color that has a hexadecimal value of #FFF5F5DC.
Bisque	The solid fill color that has a hexadecimal value of #FFFFE4C4.
Black	The solid fill color that has a hexadecimal value of #FF000000.
BlanchedAlmond	The solid fill color that has a hexadecimal value of #FFFFEBCD.
Blue	The solid fill color that has a hexadecimal value of #FF0000FF.
BlueViolet	The solid fill color that has a hexadecimal value of #FF8A2BE2.
Brown	The solid fill color that has a hexadecimal value of #FFA52A2A.
BurlyWood	The solid fill color that has a hexadecimal value of #FFDEB887.
CadetBlue	The solid fill color that has a hexadecimal value of #FF5F9EA0.
Chartreuse	The solid fill color that has a hexadecimal value of #FF7FFF00.
Chocolate	The solid fill color that has a hexadecimal value of #FFD2691E.

[values]	
Coral	The solid fill color that has a hexadecimal value of #FFFF7F50.
CornflowerBlue	The solid fill color that has a hexadecimal value of #FF6495ED.
Cornsilk	The solid fill color that has a hexadecimal value of #FFFFF8DC.
Crimson	The solid fill color that has a hexadecimal value of #FFDC143C.
Cyan	The solid fill color that has a hexadecimal value of #FF00FFFF.
DarkBlue	The solid fill color that has a hexadecimal value of #FF00008B.
DarkCyan	The solid fill color that has a hexadecimal value of #FF008B8B.
DarkGoldenrod	The solid fill color that has a hexadecimal value of #FFB8860B.
DarkGray	The solid fill color that has a hexadecimal value of #FFA9A9A9.
DarkGreen	The solid fill color that has a hexadecimal value of #FF006400.
DarkKhaki	The solid fill color that has a hexadecimal value of #FFBDB76B.
DarkMagenta	The solid fill color that has a hexadecimal value of #FF8B008B.
DarkOliveGreen	The solid fill color that has a hexadecimal value of #FF556B2F.
DarkOrange	The solid fill color that has a hexadecimal value of #FFFF8C00.
DarkOrchid	The solid fill color that has a hexadecimal value of #FF9932CC.
DarkRed	The solid fill color that has a hexadecimal value of #FF8B0000.
DarkSalmon	The solid fill color that has a hexadecimal value of #FFE9967A.
DarkSeaGreen	The solid fill color that has a hexadecimal value of #FF8FBC8F.
DarkSlateBlue	The solid fill color that has a hexadecimal value of #FF483D8B.
DarkSlateGray	The solid fill color that has a hexadecimal value

[values]	
	of #FF2F4F4F.
DarkTurquoise	The solid fill color that has a hexadecimal value of #FF00CED1.
DarkViolet	The solid fill color that has a hexadecimal value of #FF9400D3.
DeepPink	The solid fill color that has a hexadecimal value of #FFFF1493.
DeepSkyBlue	The solid fill color that has a hexadecimal value of #FF00BFFF.
DimGray	The solid fill color that has a hexadecimal value of #FF696969.
DodgerBlue	The solid fill color that has a hexadecimal value of #FF1E90FF.
Firebrick	The solid fill color that has a hexadecimal value of #FFB22222.
FloralWhite	The solid fill color that has a hexadecimal value of #FFFFFAF0.
ForestGreen	The solid fill color that has a hexadecimal value of #FF228B22.
Fuchsia	The solid fill color that has a hexadecimal value of #FFFF00FF.
Gainsboro	The solid fill color that has a hexadecimal value of #FFDCDCDC.
GhostWhite	The solid fill color that has a hexadecimal value of #FFF8F8FF.
Gold	The solid fill color that has a hexadecimal value of #FFFFD700.
Goldenrod	The solid fill color that has a hexadecimal value of #FFDAA520.
Gray	The solid fill color that has a hexadecimal value of #FF808080.
Green	The solid fill color that has a hexadecimal value of #FF008000.
GreenYellow	The solid fill color that has a hexadecimal value of #FFADFF2F.
Honeydew	The solid fill color that has a hexadecimal value of #FFF0FFF0.
HotPink	The solid fill color that has a hexadecimal value of #FFFF69B4.

[values]	
IndianRed	The solid fill color that has a hexadecimal value of #FFCD5C5C.
Indigo	The solid fill color that has a hexadecimal value of #FF4B0082.
Ivory	The solid fill color that has a hexadecimal value of #FFFFFFF0.
Khaki	The solid fill color that has a hexadecimal value of #FFF0E68C.
Lavender	The solid fill color that has a hexadecimal value of #FFE6E6FA.
LavenderBlush	The solid fill color that has a hexadecimal value of #FFFFF0F5.
LawnGreen	The solid fill color that has a hexadecimal value of #FF7CFC00.
LemonChiffon	The solid fill color that has a hexadecimal value of #FFFFFACD.
LightBlue	The solid fill color that has a hexadecimal value of #FFADD8E6.
LightCoral	The solid fill color that has a hexadecimal value of #FFF08080.
LightCyan	The solid fill color that has a hexadecimal value of #FFE0FFFF.
LightGoldenrodYellow	The solid fill color that has a hexadecimal value of #FFFAFAD2.
LightGray	The solid fill color that has a hexadecimal value of #FFD3D3D3.
LightGreen	The solid fill color that has a hexadecimal value of #FF90EE90.
LightPink	The solid fill color that has a hexadecimal value of #FFFFB6C1.
LightSalmon	The solid fill color that has a hexadecimal value of #FFFFA07A.
LightSeaGreen	The solid fill color that has a hexadecimal value of #FF20B2AA.
LightSkyBlue	The solid fill color that has a hexadecimal value of #FF87CEFA.
LightSlateGray	The solid fill color that has a hexadecimal value of #FF778899.
LightSteelBlue	The solid fill color that has a hexadecimal value

[values]	
	of #FFB0C4DE.
LightYellow	The solid fill color that has a hexadecimal value of #FFFFFFE0.
Lime	The solid fill color that has a hexadecimal value of #FF00FF00.
LimeGreen	The solid fill color that has a hexadecimal value of #FF32CD32.
Linen	The solid fill color that has a hexadecimal value of #FFFAF0E6.
Magenta	The solid fill color that has a hexadecimal value of #FFFF00FF.
Maroon	The solid fill color that has a hexadecimal value of #FF800000.
MediumAquamarine	The solid fill color that has a hexadecimal value of #FF66CDAA.
MediumBlue	The solid fill color that has a hexadecimal value of #FF0000CD.
MediumOrchid	The solid fill color that has a hexadecimal value of #FFBA55D3.
MediumPurple	The solid fill color that has a hexadecimal value of #FF9370DB.
MediumSeaGreen	The solid fill color that has a hexadecimal value of #FF3CB371.
MediumSlateBlue	The solid fill color that has a hexadecimal value of #FF7B68EE.
MediumSpringGreen	The solid fill color that has a hexadecimal value of #FF00FA9A.
MediumTurquoise	The solid fill color that has a hexadecimal value of #FF48D1CC.
MediumVioletRed	The solid fill color that has a hexadecimal value of #FFC71585.
MidnightBlue	The solid fill color that has a hexadecimal value of #FF191970.
MintCream	The solid fill color that has a hexadecimal value of #FFF5FFFA.
MistyRose	The solid fill color that has a hexadecimal value of #FFFFE4E1.
Moccasin	The solid fill color that has a hexadecimal value of #FFFFE4B5.

[values]	
NavajoWhite	The solid fill color that has a hexadecimal value of #FFFFDEAD.
Navy	The solid fill color that has a hexadecimal value of #FF000080.
OldLace	The solid fill color that has a hexadecimal value of #FFFDF5E6.
Olive	The solid fill color that has a hexadecimal value of #FF808000.
OliveDrab	The solid fill color that has a hexadecimal value of #FF6B8E23.
Orange	The solid fill color that has a hexadecimal value of #FFFFA500.
OrangeRed	The solid fill color that has a hexadecimal value of #FFFF4500.
Orchid	The solid fill color that has a hexadecimal value of #FFDA70D6.
PaleGoldenrod	The solid fill color that has a hexadecimal value of #FFEEE8AA.
PaleGreen	The solid fill color that has a hexadecimal value of #FF98FB98.
PaleTurquoise	The solid fill color that has a hexadecimal value of #FFAFEEEE.
PaleVioletRed	The solid fill color that has a hexadecimal value of #FFDB7093.
PapayaWhip	The solid fill color that has a hexadecimal value of #FFFFEFD5.
PeachPuff	The solid fill color that has a hexadecimal value of #FFFFDAB9.
Peru	The solid fill color that has a hexadecimal value of #FFCD853F.
Pink	The solid fill color that has a hexadecimal value of #FFFFC0CB.
Plum	The solid fill color that has a hexadecimal value of #FFDDA0DD.
PowderBlue	The solid fill color that has a hexadecimal value of #FFB0E0E6.
Purple	The solid fill color that has a hexadecimal value of #FF800080.
Red	The solid fill color that has a hexadecimal value

[values]	
	of #FFFF0000.
RosyBrown	The solid fill color that has a hexadecimal value of #FFBC8F8F.
RoyalBlue	The solid fill color that has a hexadecimal value of #FF4169E1.
SaddleBrown	The solid fill color that has a hexadecimal value of #FF8B4513.
Salmon	The solid fill color that has a hexadecimal value of #FFFA8072.
SandyBrown	The solid fill color that has a hexadecimal value of #FFF4A460.
SeaGreen	The solid fill color that has a hexadecimal value of #FF2E8B57.
SeaShell	The solid fill color that has a hexadecimal value of #FFFFF5EE.
Sienna	The solid fill color that has a hexadecimal value of #FFA0522D.
Silver	The solid fill color that has a hexadecimal value of #FFC0C0C0.
SkyBlue	The solid fill color that has a hexadecimal value of #FF87CEEB.
SlateBlue	The solid fill color that has a hexadecimal value of #FF6A5ACD.
SlateGray	The solid fill color that has a hexadecimal value of #FF708090.
Snow	The solid fill color that has a hexadecimal value of #FFFFFAFA.
SpringGreen	The solid fill color that has a hexadecimal value of #FF00FF7F.
SteelBlue	The solid fill color that has a hexadecimal value of #FF4682B4.
Tan	The solid fill color that has a hexadecimal value of #FFD2B48C.
Teal	The solid fill color that has a hexadecimal value of #FF008080.
Thistle	The solid fill color that has a hexadecimal value of #FFD8BFD8.
Tomato	The solid fill color that has a hexadecimal value of #FFFF6347.

[values]	
Transparent	The solid fill color that has a hexadecimal value of #00FFFFFF.
Turquoise	The solid fill color that has a hexadecimal value of #FF40E0D0.
Violet	The solid fill color that has a hexadecimal value of #FFEE82EE.
Wheat	The solid fill color that has a hexadecimal value of #FFF5DEB3.
White	The solid fill color that has a hexadecimal value of #FFFFFFFF.
WhiteSmoke	The solid fill color that has a hexadecimal value of #FFF5F5F5.
Yellow	The solid fill color that has a hexadecimal value of #FFFFFF00.
YellowGreen	The solid fill color that has a hexadecimal value of #FF9ACD32.
[patterns]	
#[\dA-F]{3}	An RGB value in the sRGB color space with one hexadecimal digit per channel. Examples: #F00 Red: 100%, Green 0%, Blue 0% #fb0 Red: 100%, Green 73%, Blue 0% #FFF Red: 100%, Green 100%, Blue 100%
#[\dA-F]{4}	An ARGB value in the sRGB color space with one hexadecimal digit per channel. Examples: #FF00 Alpha: 100% Red: 100%, Green 0%, Blue 0% #8fb0 Alpha: 53% Red: 100%, Green 73%, Blue 0% #4FFF Alpha 27% Red: 100%, Green 100%, Blue 100%
#[\dA-F]{6}	An RGB value in the sRGB color space with two hexadecimal digits per channel. Examples: #FF0000 Red: 100%, Green 0%, Blue 0% #ff3300 Red: 100%, Green 20%, Blue 0% #FFFFFF Red: 100%, Green 100%, Blue 100%
#[\dA-F]{8}	An ARGB value in the sRGB color space with two hexadecimal digits per channel. Examples: #FFFF0000 Alpha: 100% Red: 100%, Green 0%, Blue 0% #80ffbf00 Alpha: 50% Red: 100%, Green 75%, Blue 0% #40FFFFFF Alpha 25% Red: 100%, Green 100%, Blue 100%
sc#\s*(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){2,3}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	An RGB or ARGB value in the scRGB color space, with each channel specified as a decimal floating point number, separated by either commas or whitespace. Examples: sc# 1 0 0 Red: 100%, Green 0%, Blue 0% sc# 1 0.75, 0 Red: 100%, Green 75%, Blue 0% sc# 2.5E-1 1

<table>
<tr><td>[values]</td><td></td></tr>
<tr><td></td><td>0 0 Alpha: 25%, Red: 100%, Green 0%, Blue 0% sc# 1.5,0,-0.5 Red: 150%, Green 0%, Blue -50%</td></tr>
<tr><td>**[is case sensitive]**</td><td>True</td></tr>
<tr><td>**ContextColor\s+[^\s]*\s*(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){3,8}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)**</td><td>A color in a color space identified by a URI; the color space URI is followed by a sequence of decimal floating point values separated by whitespace and/or commas. The first is an alpha value, and it is followed by the color-space-specific channel values (of which there may be from 3 to 8 inclusive). Examples: ContextColor ../color/sRGB.icm 1 1 0 0</td></tr>
<tr><td>**[is case sensitive]**</td><td>True</td></tr>
</table>

CacheModeSyntax

[values]	
BitmapCache	Caches the visual content as a bitmap.

CaptureStateSyntax (4)

[values]	
Failed	The capture failed, or no operation has yet been attempted.
Started	Start has been called and the capture is still active.
Stopped	Stop has been called and the capture is complete.

ClickModeSyntax

[values]	
Hover	Specifies that the Click event should be raised when the mouse pointer moves over the control.
Press	Specifies that the Click event should be raised when the mouse button is pressed and the mouse pointer is over the control. If you are using the keyboard, specifies that the Click event should be raised when the SPACEBAR or ENTER is pressed and the control has keyboard focus.
Release	Specifies that the Click event should be raised when the left mouse button is pressed and released, and the mouse pointer is over the control. If you are using the keyboard, specifies that the Click event should be raised when the SPACEBAR or ENTER key is pressed and released, and the control has keyboard focus.

ClockStateSyntax

[values]	
Active	The current animation changes in direct relation to that of its parent.

[values]	
Filling	The animation continues and does not change in relation to that of its parent.
Stopped	The animation is stopped.

ColorInterpolationModeSyntax

[values]	
ScRgbLinearInterpolation	Colors are interpolated in the scRGB color space
SRgbLinearInterpolation	Colors are interpolated in the sRGB color space

ColorSyntax

[values]	
AliceBlue	The solid fill color that has a hexadecimal value of #FFF0F8FF.
AntiqueWhite	The solid fill color that has a hexadecimal value of #FFFAEBD7.
Aqua	The solid fill color that has a hexadecimal value of #FF00FFFF.
Aquamarine	The solid fill color that has a hexadecimal value of #FF7FFFD4.
Azure	The solid fill color that has a hexadecimal value of #FFF0FFFF.
Beige	The solid fill color that has a hexadecimal value of #FFF5F5DC.
Bisque	The solid fill color that has a hexadecimal value of #FFFFE4C4.
Black	The solid fill color that has a hexadecimal value of #FF000000.
BlanchedAlmond	The solid fill color that has a hexadecimal value of #FFFFEBCD.
Blue	The solid fill color that has a hexadecimal value of #FF0000FF.
BlueViolet	The solid fill color that has a hexadecimal value of #FF8A2BE2.
Brown	The solid fill color that has a hexadecimal value of #FFA52A2A.
BurlyWood	The solid fill color that has a hexadecimal value of #FFDEB887.
CadetBlue	The solid fill color that has a hexadecimal value

[values]	
	of #FF5F9EA0.
Chartreuse	The solid fill color that has a hexadecimal value of #FF7FFF00.
Chocolate	The solid fill color that has a hexadecimal value of #FFD2691E.
Coral	The solid fill color that has a hexadecimal value of #FFFF7F50.
CornflowerBlue	The solid fill color that has a hexadecimal value of #FF6495ED.
Cornsilk	The solid fill color that has a hexadecimal value of #FFFFF8DC.
Crimson	The solid fill color that has a hexadecimal value of #FFDC143C.
Cyan	The solid fill color that has a hexadecimal value of #FF00FFFF.
DarkBlue	The solid fill color that has a hexadecimal value of #FF00008B.
DarkCyan	The solid fill color that has a hexadecimal value of #FF008B8B.
DarkGoldenrod	The solid fill color that has a hexadecimal value of #FFB8860B.
DarkGray	The solid fill color that has a hexadecimal value of #FFA9A9A9.
DarkGreen	The solid fill color that has a hexadecimal value of #FF006400.
DarkKhaki	The solid fill color that has a hexadecimal value of #FFBDB76B.
DarkMagenta	The solid fill color that has a hexadecimal value of #FF8B008B.
DarkOliveGreen	The solid fill color that has a hexadecimal value of #FF556B2F.
DarkOrange	The solid fill color that has a hexadecimal value of #FFFF8C00.
DarkOrchid	The solid fill color that has a hexadecimal value of #FF9932CC.
DarkRed	The solid fill color that has a hexadecimal value of #FF8B0000.
DarkSalmon	The solid fill color that has a hexadecimal value of #FFE9967A.

[values]	
DarkSeaGreen	The solid fill color that has a hexadecimal value of #FF8FBC8F.
DarkSlateBlue	The solid fill color that has a hexadecimal value of #FF483D8B.
DarkSlateGray	The solid fill color that has a hexadecimal value of #FF2F4F4F.
DarkTurquoise	The solid fill color that has a hexadecimal value of #FF00CED1.
DarkViolet	The solid fill color that has a hexadecimal value of #FF9400D3.
DeepPink	The solid fill color that has a hexadecimal value of #FFFF1493.
DeepSkyBlue	The solid fill color that has a hexadecimal value of #FF00BFFF.
DimGray	The solid fill color that has a hexadecimal value of #FF696969.
DodgerBlue	The solid fill color that has a hexadecimal value of #FF1E90FF.
Firebrick	The solid fill color that has a hexadecimal value of #FFB22222.
FloralWhite	The solid fill color that has a hexadecimal value of #FFFFFAF0.
ForestGreen	The solid fill color that has a hexadecimal value of #FF228B22.
Fuchsia	The solid fill color that has a hexadecimal value of #FFFF00FF.
Gainsboro	The solid fill color that has a hexadecimal value of #FFDCDCDC.
GhostWhite	The solid fill color that has a hexadecimal value of #FFF8F8FF.
Gold	The solid fill color that has a hexadecimal value of #FFFFD700.
Goldenrod	The solid fill color that has a hexadecimal value of #FFDAA520.
Gray	The solid fill color that has a hexadecimal value of #FF808080.
Green	The solid fill color that has a hexadecimal value of #FF008000.
GreenYellow	The solid fill color that has a hexadecimal value

[values]	
	of #FFADFF2F.
Honeydew	The solid fill color that has a hexadecimal value of #FFF0FFF0.
HotPink	The solid fill color that has a hexadecimal value of #FFFF69B4.
IndianRed	The solid fill color that has a hexadecimal value of #FFCD5C5C.
Indigo	The solid fill color that has a hexadecimal value of #FF4B0082.
Ivory	The solid fill color that has a hexadecimal value of #FFFFFFF0.
Khaki	The solid fill color that has a hexadecimal value of #FFF0E68C.
Lavender	The solid fill color that has a hexadecimal value of #FFE6E6FA.
LavenderBlush	The solid fill color that has a hexadecimal value of #FFFFF0F5.
LawnGreen	The solid fill color that has a hexadecimal value of #FF7CFC00.
LemonChiffon	The solid fill color that has a hexadecimal value of #FFFFFACD.
LightBlue	The solid fill color that has a hexadecimal value of #FFADD8E6.
LightCoral	The solid fill color that has a hexadecimal value of #FFF08080.
LightCyan	The solid fill color that has a hexadecimal value of #FFE0FFFF.
LightGoldenrodYellow	The solid fill color that has a hexadecimal value of #FFFAFAD2.
LightGray	The solid fill color that has a hexadecimal value of #FFD3D3D3.
LightGreen	The solid fill color that has a hexadecimal value of #FF90EE90.
LightPink	The solid fill color that has a hexadecimal value of #FFFFB6C1.
LightSalmon	The solid fill color that has a hexadecimal value of #FFFFA07A.
LightSeaGreen	The solid fill color that has a hexadecimal value of #FF20B2AA.

[values]	
LightSkyBlue	The solid fill color that has a hexadecimal value of #FF87CEFA.
LightSlateGray	The solid fill color that has a hexadecimal value of #FF778899.
LightSteelBlue	The solid fill color that has a hexadecimal value of #FFB0C4DE.
LightYellow	The solid fill color that has a hexadecimal value of #FFFFFFE0.
Lime	The solid fill color that has a hexadecimal value of #FF00FF00.
LimeGreen	The solid fill color that has a hexadecimal value of #FF32CD32.
Linen	The solid fill color that has a hexadecimal value of #FFFAF0E6.
Magenta	The solid fill color that has a hexadecimal value of #FFFF00FF.
Maroon	The solid fill color that has a hexadecimal value of #FF800000.
MediumAquamarine	The solid fill color that has a hexadecimal value of #FF66CDAA.
MediumBlue	The solid fill color that has a hexadecimal value of #FF0000CD.
MediumOrchid	The solid fill color that has a hexadecimal value of #FFBA55D3.
MediumPurple	The solid fill color that has a hexadecimal value of #FF9370DB.
MediumSeaGreen	The solid fill color that has a hexadecimal value of #FF3CB371.
MediumSlateBlue	The solid fill color that has a hexadecimal value of #FF7B68EE.
MediumSpringGreen	The solid fill color that has a hexadecimal value of #FF00FA9A.
MediumTurquoise	The solid fill color that has a hexadecimal value of #FF48D1CC.
MediumVioletRed	The solid fill color that has a hexadecimal value of #FFC71585.
MidnightBlue	The solid fill color that has a hexadecimal value of #FF191970.
MintCream	The solid fill color that has a hexadecimal value

[values]	
	of #FFF5FFFA.
MistyRose	The solid fill color that has a hexadecimal value of #FFFFE4E1.
Moccasin	The solid fill color that has a hexadecimal value of #FFFFE4B5.
NavajoWhite	The solid fill color that has a hexadecimal value of #FFFFDEAD.
Navy	The solid fill color that has a hexadecimal value of #FF000080.
OldLace	The solid fill color that has a hexadecimal value of #FFFDF5E6.
Olive	The solid fill color that has a hexadecimal value of #FF808000.
OliveDrab	The solid fill color that has a hexadecimal value of #FF6B8E23.
Orange	The solid fill color that has a hexadecimal value of #FFFFA500.
OrangeRed	The solid fill color that has a hexadecimal value of #FFFF4500.
Orchid	The solid fill color that has a hexadecimal value of #FFDA70D6.
PaleGoldenrod	The solid fill color that has a hexadecimal value of #FFEEE8AA.
PaleGreen	The solid fill color that has a hexadecimal value of #FF98FB98.
PaleTurquoise	The solid fill color that has a hexadecimal value of #FFAFEEEE.
PaleVioletRed	The solid fill color that has a hexadecimal value of #FFDB7093.
PapayaWhip	The solid fill color that has a hexadecimal value of #FFFFEFD5.
PeachPuff	The solid fill color that has a hexadecimal value of #FFFFDAB9.
Peru	The solid fill color that has a hexadecimal value of #FFCD853F.
Pink	The solid fill color that has a hexadecimal value of #FFFFC0CB.
Plum	The solid fill color that has a hexadecimal value of #FFDDA0DD.

[values]	
PowderBlue	The solid fill color that has a hexadecimal value of #FFB0E0E6.
Purple	The solid fill color that has a hexadecimal value of #FF800080.
Red	The solid fill color that has a hexadecimal value of #FFFF0000.
RosyBrown	The solid fill color that has a hexadecimal value of #FFBC8F8F.
RoyalBlue	The solid fill color that has a hexadecimal value of #FF4169E1.
SaddleBrown	The solid fill color that has a hexadecimal value of #FF8B4513.
Salmon	The solid fill color that has a hexadecimal value of #FFFA8072.
SandyBrown	The solid fill color that has a hexadecimal value of #FFF4A460.
SeaGreen	The solid fill color that has a hexadecimal value of #FF2E8B57.
SeaShell	The solid fill color that has a hexadecimal value of #FFFFF5EE.
Sienna	The solid fill color that has a hexadecimal value of #FFA0522D.
Silver	The solid fill color that has a hexadecimal value of #FFC0C0C0.
SkyBlue	The solid fill color that has a hexadecimal value of #FF87CEEB.
SlateBlue	The solid fill color that has a hexadecimal value of #FF6A5ACD.
SlateGray	The solid fill color that has a hexadecimal value of #FF708090.
Snow	The solid fill color that has a hexadecimal value of #FFFFFAFA.
SpringGreen	The solid fill color that has a hexadecimal value of #FF00FF7F.
SteelBlue	The solid fill color that has a hexadecimal value of #FF4682B4.
Tan	The solid fill color that has a hexadecimal value of #FFD2B48C.
Teal	The solid fill color that has a hexadecimal value

[values]	
	of #FF008080.
Thistle	The solid fill color that has a hexadecimal value of #FFD8BFD8.
Tomato	The solid fill color that has a hexadecimal value of #FFFF6347.
Transparent	The solid fill color that has a hexadecimal value of #00FFFFFF.
Turquoise	The solid fill color that has a hexadecimal value of #FF40E0D0.
Violet	The solid fill color that has a hexadecimal value of #FFEE82EE.
Wheat	The solid fill color that has a hexadecimal value of #FFF5DEB3.
White	The solid fill color that has a hexadecimal value of #FFFFFFFF.
WhiteSmoke	The solid fill color that has a hexadecimal value of #FFF5F5F5.
Yellow	The solid fill color that has a hexadecimal value of #FFFFFF00.
YellowGreen	The solid fill color that has a hexadecimal value of #FF9ACD32.
[patterns]	
#[\dA-F]{3}	An RGB value in the sRGB color space with one hexadecimal digit per channel. Examples: #F00 Red: 100%, Green 0%, Blue 0% #fb0 Red: 100%, Green 73%, Blue 0% #FFF Red: 100%, Green 100%, Blue 100%
#[\dA-F]{4}	An ARGB value in the sRGB color space with one hexadecimal digit per channel. Examples: #FF00 Alpha: 100% Red: 100%, Green 0%, Blue 0% #8fb0 Alpha: 53% Red: 100%, Green 73%, Blue 0% #4FFF Alpha 27% Red: 100%, Green 100%, Blue 100%
#[\dA-F]{6}	An RGB value in the sRGB color space with two hexadecimal digits per channel. Examples: #FF0000 Red: 100%, Green 0%, Blue 0% #ff3300 Red: 100%, Green 20%, Blue 0% #FFFFFF Red: 100%, Green 100%, Blue 100%
#[\dA-F]{8}	An ARGB value in the sRGB color space with two hexadecimal digits per channel. Examples: #FFFF0000 Alpha: 100% Red: 100%, Green 0%, Blue 0% #80ffbf00 Alpha: 50% Red: 100%, Green 75%, Blue 0% #40FFFFFF Alpha

[values]	
	25% Red: 100%, Green 100%, Blue 100%
sc#\s*(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){2,3}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	An RGB or ARGB value in the scRGB color space, with each channel specified as a decimal floating point number, separated by either commas or whitespace. Examples: sc# 1 0 0 Red: 100%, Green 0%, Blue 0% sc# 1 0.75, 0 Red: 100%, Green 75%, Blue 0% sc# 2.5E-1 1 0 0 Alpha: 25%, Red: 100%, Green 0%, Blue 0% sc# 1.5,0,-0.5 Red: 150%, Green 0%, Blue -50%
[is case sensitive]	True
ContextColor\s+[^\s]*\s*(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){3,8}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	A color in a color space identified by a URI; the color space URI is followed by a sequence of decimal floating point values separated by whitespace and/or commas. The first is an alpha value, and it is followed by the color-space-specific channel values (of which there may be from 3 to 8 inclusive). Examples: ContextColor ../color/sRGB.icm 1 1 0 0
[is case sensitive]	True

ContentKeyTypeSyntax (4)

[values]	
Aes128Bit	Content key type used for Root licenses (AES-ECB).
Cocktail	Content key type used for Cocktail in the PlayReady ecosystem.

CornerRadiusSyntax

[patterns]	
(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){3}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	Four whitespace and/or comma separated decimal floating point values specifying the radius of curvature for the top left, top right, bottom right, and bottom left corners of a shape.
[is case sensitive]	True
([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	A single decimal floating point value indicating the radius of curvature for a corner or some corners.
[is case sensitive]	True

CrossDomainAccessSyntax

[values]	

[values]	
NoAccess	Cross-domain callers have no access to the Silverlight application.
ScriptableOnly	Cross-domain callers have script access to the Silverlight application.

CursorsSyntax

[values]	
Arrow	Represents an Arrow Cursor.
Eraser	Represents an Eraser Cursor.
Hand	Represents a Hand Cursor.
IBeam	Represents an IBeam Cursor, which is typically used to show where the text cursor appears when the mouse is clicked.
None	Represents a special Cursor that is invisible.
SizeNESW	Represents a SizeNESW Cursor.
SizeNS	Represents a SizeNS Cursor.
SizeNWSE	Represents a SizeNWSE Cursor.
SizeWE	Represents a SizeWE Cursor.
Stylus	Represents a Stylus Cursor.
Wait	Represents a WaitCursor.

CursorSyntax

[values]	
AppStarting	The Cursor that appears when an application is starting.
Arrow	The Arrow Cursor.
ArrowCD	The arrow with a compact disk Cursor.
Cross	The crosshair Cursor.
Hand	A hand Cursor.
Help	A help Cursor which is a combination of an arrow and a question mark.
IBeam	An I-beam Cursor, which is used to show where the text cursor appears when the mouse is clicked.
No	A Cursor with which indicates that a particular region is invalid for a given operation.
None	A special cursor that is invisible.
Pen	A pen Cursor.

[values]	
ScrollAll	The scroll all Cursor.
ScrollE	The scroll east Cursor.
ScrollN	The scroll north Cursor.
ScrollNE	The scroll northeast cursor.
ScrollNS	The scroll north/south cursor.
ScrollNW	A scroll northwest cursor.
ScrollS	The scroll south Cursor.
ScrollSE	A south/east scrolling Cursor.
ScrollSW	The scroll southwest Cursor.
ScrollW	The scroll west Cursor.
ScrollWE	A west/east scrolling Cursor.
SizeAll	A four-headed sizing Cursor, which consists of four joined arrows that point north, south, east, and west.
SizeNESW	A two-headed northeast/southwest sizing Cursor.
SizeNS	A two-headed north/south sizing Cursor.
SizeNWSE	A two-headed northwest/southeast sizing Cursor.
SizeWE	A two-headed west/east sizing Cursor.
UpArrow	An up arrow Cursor, which is typically used to identify an insertion point.
Wait	Specifies a wait (or hourglass) Cursor.
[patterns]	
.*\.((CUR)|(ANI))	Any string ending in either ".cur" or ".ani"

DockPositionSyntax

[values]	
Bottom	Indicates that the UI automation element is docked along the bottom edge of the docking container.
Fill	Indicates that the UI automation element is docked along all edges of the docking container and fills all available space within the container.
Left	Indicates that the UI automation element is docked along the left edge of the docking container.
None	Indicates that the UI automation element is not docked to any edge of the docking container
Right	Indicates that the UI automation element is docked along the right edge of the docking

[values]	
	container.
Top	Indicates that the UI automation element is docked along the top edge of the docking container.

DoubleCollectionSyntax

[patterns]	
([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)(((\s*,\s*)|\s+)([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?))*	One or more whitespace and/or comma separated decimal floating point values.
[is case sensitive]	True

DurationSyntax

[values]	
Automatic	
Forever	
[patterns]	
(\d+\.)?\d\d?:\d\d?:((\d\d?)|(\d?\d?\.\d*))	The string may optionally begin with a decimal number and a dot. When present, this optional part indicates the number of days. The string always contains three parts separated by colons. The first two of these are one or two digit numbers specifying hours and minutes. The third part indicates the number of seconds and can be a two-digit integer, or a floating point number with 0, 1, or 2 digits before the point, and any number of digits after the point. Examples: 0:0:1 One second 1.0:0:0 One day 0:0:0.5 Half a second 0:0:.5 Half a second 2.5:2:22 Two days, five hours, two minutes, 22 seconds
[is case sensitive]	True
\d+	A decimal number specifying the number of days. Examples: 1 One day 10 Ten days
[is case sensitive]	True

EasingModeSyntax

[values]	
EaseIn	Interpolation follows the mathematical formula associated with the easing function.
EaseInOut	Interpolation uses EaseIn for the first half of the animation and EaseOut for the second half.
EaseOut	Interpolation follows 100% interpolation minus the output of the formula associated with

[values]	
	the easing function.

ElevatedPermissionsSyntax (4)

[values]	
NotRequired	Elevated permissions are not required to run the application outside the browser.
Required	Elevated permissions are required to run the application outside the browser.

ExpandCollapseStateSyntax

[values]	
Collapsed	No child nodes, controls, or content of the UI automation element are displayed.
Expanded	All child nodes, controls or content of the UI automation element are displayed.
LeafNode	The UI automation element has no child nodes, controls, or content to display.
PartiallyExpanded	Some, but not all, child nodes, controls, or content of the UI automation element are displayed.

FillBehaviorSyntax

[values]	
HoldEnd	After it reaches the end of its active period, the timeline holds its progress until the end of its parent's active and hold periods.
Stop	The timeline stops if it is outside its active period while its parent is inside its active period.

FillRuleSyntax

[values]	
EvenOdd	Rule that determines whether a point is in the fill region by drawing a ray from that point to infinity in any direction and counting the number of path segments within the given shape that the ray crosses. If this number is odd, the point is inside; if even, the point is outside.
Nonzero	Rule that determines whether a point is in the fill region of the path by drawing a ray from that point to infinity in any direction and then examining the places where a segment of the shape crosses the ray. Starting with a count of 0, add one each time a segment crosses the ray from left to right and subtract one each time a path segment crosses the ray from right to left. After counting the crossings, if the result is 0 then the point is outside the path. Otherwise, it is inside.

FlowDirectionSyntax (4)

[values]	
LeftToRight	Indicates that content should flow from left to right.

[values]	
RightToLeft	Indicates that content should flow from right to left.

FontFamilySyntax

[patterns]	
.*	A sequence of comma-separated font family names. Each name can optionally start with a string indicating the location of the font file. This optional location specifier is indicated by a # symbol - the part before the hash is the location and the part after the hash is the family name. The absence of a # indicates that only the family name is specified. (The regular expression does not reflect this, because there are no restrictions on what text appears as the font name other than that it must not contain a '#' or a ',' and since those are both allowed as delimiters, there are no syntactic limits on the string. Of course whether the string is meaningful in practice depends on whether the specified font is available.)
[is case sensitive]	True

FontStretchSyntax

[values]	
Condensed	Specifies a condensed FontStretch.
[trim whitespace]	False
Expanded	Specifies an expanded FontStretch.
[trim whitespace]	False
ExtraCondensed	Specifies an extra-condensed FontStretch.
[trim whitespace]	False
ExtraExpanded	Specifies an extra-expanded FontStretch.
[trim whitespace]	False
Medium	Specifies a medium FontStretch.
[trim whitespace]	False
Normal	Specifies a normal FontStretch.
[trim whitespace]	False
SemiCondensed	Specifies a semi-condensed FontStretch.
[trim	False

[values]	
whitespace]	
SemiExpanded	Specifies a semi-expanded FontStretch.
[trim whitespace]	False
UltraCondensed	Specifies an ultra-condensed FontStretch.
[trim whitespace]	False
UltraExpanded	Specifies an ultra-expanded FontStretch.
[trim whitespace]	False
[patterns]	
\+?0*[1-9]	A single digit in the range 1-9, optionally preceded by a + sign and/or any number of leading zeros.
[is case sensitive]	True

FontWeightSyntax

[values]	
Black	Specifies a "Black" font weight.
[trim whitespace]	False
Bold	Specifies a "Bold" font weight.
[trim whitespace]	False
DemiBold	Specifies a "Demi-bold" font weight.
[trim whitespace]	False
ExtraBlack	Specifies an "Extra-black" font weight.
[trim whitespace]	False
ExtraBold	Specifies an "Extra-bold" font weight.
[trim whitespace]	False
ExtraLight	Specifies an "Extra-light" font weight.
[trim whitespace]	False
Heavy	Specifies a "Heavy" font weight.
[trim whitespace]	False
Light	Specifies a "Light" font weight.

[values]	
[trim whitespace]	False
Medium	Specifies a "Medium" font weight.
[trim whitespace]	False
Normal	Specifies a "Normal" font weight.
[trim whitespace]	False
Regular	Specifies a "Regular" font weight.
[trim whitespace]	False
SemiBold	Specifies a "Semi-bold" font weight.
[trim whitespace]	False
Thin	Specifies a "Thin" font weight.
[trim whitespace]	False
UltraBlack	Specifies an "Ultra-black" font weight.
[trim whitespace]	False
UltraBold	Specifies an "Ultra-bold" font weight.
[trim whitespace]	False
UltraLight	Specifies an "Ultra-light" font weight.
[trim whitespace]	False
[patterns]	
\+?\d*	An integer numeric value. (This should be in the range 1-999.)
[is case sensitive]	True

GeneratorDirectionSyntax

[values]	
Backward	Specifies to generate items in a backward direction.
Forward	Specifies to generate items in a forward direction.

GeometrySyntax

<table>
<tr><td>[patterns]</td><td></td></tr>
<tr><td>(F\s*[01])?(\s*[mMlLhHvVcCsSqQtTaAzZ]\s*((([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+))*([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?))?)*</td><td>Optionally begins with a fill rule specifier: an uppercase F followed by a 0 or a 1, with optional whitespace between the F and</td></tr>
</table>

[patterns]	
	the digit. Remainder of string consists of a sequence of single letter commands, each followed by a sequence of comma and/or whitespace-separated floating point decimal values.
[is case sensitive]	True

GradientSpreadMethodSyntax

[values]	
Pad	The color values at the ends of the gradient vector fill the remaining space.
Reflect	The gradient is repeated in the reverse direction until the space is filled.
Repeat	The gradient is repeated in the original direction until the space is filled.

GridLengthSyntax

[values]	
auto	Indicates that automatic sizing should be used
*****	Equivalent to a value of "1*"
[patterns]	
[+-]?((\d[\d,]*(\.\d*)?)|((\d[\d,]*)?\.\d+))([eE][+-]?\d+)? (\s*(*|PX|IN|CM|PT))?	A decimal floating point number, optionally followed by either '*', 'in', 'cm', 'pt', or 'px'

GridUnitTypeSyntax

[values]	
Auto	The size is determined by the size properties of the content object.
Pixel	The value is expressed in pixels.
Star	The value is expressed as a weighted proportion of available space.

HorizontalAlignmentSyntax

[values]	
Center	An element aligned to the center of the layout slot for the parent element.
Left	An element aligned to the left of the layout slot for the parent element.
Right	An element aligned to the right of the layout slot for the parent element.

[values]	
Stretch	An element stretched to fill the entire layout slot of the parent element.

ImeConversionModeValuesSyntax (4)

[values]	
Alphanumeric	The input method uses alphanumeric conversion mode.
CharCode	The input method uses character code conversion mode.
DoNotCare	The input method does not care what input conversion method is used; the actual conversion method is indeterminate.
Eudc	The input method uses EUDC (end user defined character) conversion mode.
Fixed	The input method uses fixed conversion mode.
FullShape	The input method uses full-shape conversion mode.
Katakana	The input method uses Katakana conversion mode.
Native	The input method uses a native character (Hiragana, Hangul, Chinese) conversion mode.
NoConversion	The input method will not perform any input conversion.
Roman	The input method uses Roman character conversion mode.
Symbol	The input method uses symbol conversion mode.

InputMethodStateSyntax (4)

[values]	
DoNotCare	The input method is in an indeterminate state.
Off	The input method editor is off for purposes of generating device independent input.
On	The input method editor is on.

InputScopeNameValueSyntax (4)

[values]	
AddressCity	The text input pattern for a city address.
AddressCountryName	The text input pattern for the name of a country/region.
AddressCountryShortName	The text input pattern for the abbreviated name of a country/region.
AddressStateOrProvince	The text input pattern for a state or province.
AddressStreet	The text input pattern for a street address.
AlphanumericFullWidth	The text input pattern for alphanumeric full-width characters.

[values]	
AlphanumericHalfWidth	The text input pattern for alphanumeric half-width characters.
ApplicationEnd	Not supported. For internal use in Silverlight for Windows Phone.
Bopomofo	The text input pattern for the Bopomofo Mandarin Chinese phonetic transcription system.
Chat	The SIP layout for text messaging, which recognizes pre-defined abbreviations. Supported only in Silverlight for Windows Phone.
CurrencyAmount	The text input pattern for amount of currency.
CurrencyAmountAndSymbol	The text input pattern for amount and symbol of currency.
CurrencyChinese	The text input pattern for Chinese currency.
Date	The text input pattern for a calendar date.
DateDay	The text input pattern for the numeric day in a calendar date.
DateDayName	The text input pattern for the name of the day in a calendar date.
DateMonth	The text input pattern for the numeric month in a calendar date.
DateMonthName	The text input pattern for the name of the month in a calendar date.
DateYear	The text input pattern for the year in a calendar date.
Default	The default handling of input commands.
Digits	The text input pattern for digits.
EmailNameOrAddress	The SIP layout for an e-mail name or address. Supported only in Silverlight for Windows Phone.
EmailSmtpAddress	The text input pattern for a Simple Mail Transfer Protocol (SMTP) email address.
EmailUserName	The text input pattern for an email user name.
EnumString	Not supported. For internal use in Silverlight for Windows Phone.
FileName	The text input pattern for a file name.
FullFilePath	The text input pattern for the full path of a file.
Hanja	The text input pattern for Hanja characters.
Hiragana	The text input pattern for the Hiragana writing system.
KatakanaFullWidth	The text input pattern for full-width Katakana characters.
KatakanaHalfWidth	The text input pattern for half-width Katakana characters.
LogOnName	The text input pattern for a log on name.
Maps	The SIP layout for entering a map location. Supported only in Silverlight for Windows Phone.

[values]	
NameOrPhoneNumber	The SIP layout for SMS To field. Supported only in Silverlight for Windows Phone.
Number	The text input pattern for a number.
NumberFullWidth	The text input pattern for a full-width number.
OneChar	The text input pattern for one character.
Password	The text input pattern for a password.
PersonalFullName	The text input pattern for a person's full name.
PersonalGivenName	The text input pattern for a person's given name.
PersonalMiddleName	The text input pattern for a person's middle name.
PersonalNamePrefix	The text input pattern for the prefix of a person's name.
PersonalNameSuffix	The text input pattern for the suffix of a person's name.
PersonalSurname	The text input pattern for a person's surname.
PhraseList	The text input pattern for a phrase list.
PostalAddress	The text input pattern for a postal address.
PostalCode	The text input pattern for a postal code.
Private	Not supported. For internal use in Silverlight for Windows Phone.
RegularExpression	The text input pattern for a regular expression.
Search	The SIP layout for a search query. Supported only in Silverlight for Windows Phone.
Srgs	The text input pattern for the Speech Recognition Grammar Specification (SRGS).
TelephoneAreaCode	The text input pattern for a telephone area code.
TelephoneCountryCode	The text input pattern for a telephone country/region code.
TelephoneLocalNumber	The text input pattern for a telephone local number.
TelephoneNumber	The text input pattern for a telephone number.
Text	The software input panel (SIP) layout for standard text input. Supported only in Silverlight for Windows Phone.
Time	The text input pattern for the time.
TimeHour	The text input pattern for the hour of the time.
TimeMinorSec	The text input pattern for the minutes or seconds of time.
Url	The text input pattern for a Uniform Resource Locator (URL).

[values]	
Xml	The text input pattern for XML.
Yomi	Not supported. For internal use in Silverlight for Windows Phone.

InstallStateSyntax

[values]	
Installed	The application has been installed to run outside the browser.
InstallFailed	The application could not be installed to run outside the browser.
Installing	The application is in the process of being installed to run outside the browser.
NotInstalled	The application has not been installed to run outside the browser.

KeyboardNavigationModeSyntax

[values]	
Cycle	Focus returns to the first or the last keyboard navigation stop inside of a container when the first or last keyboard navigation stop is reached.
Local	Tab indexes are considered on the local subtree only inside this container.
Once	The container and all of its child elements as a whole receive focus only once.

KeySplineSyntax

[patterns]	
(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){3}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	Four decimal floating point values, separated by either a comma or whitespace.
[is case sensitive]	True

KeySyntax

[values]	
A	The A key.
Add	The + (ADD) key.
Alt	The ALT key.
B	The B key.
Back	The BACKSPACE key.
C	The C key.
CapsLock	The CAPSLOCK key.

[values]	
Ctrl	The CTRL (control) key.
D	The D key.
D0	The 0 (zero) key.
D1	The 1 key.
D2	The 2 key.
D3	The 3 key.
D4	The 4 key.
D5	The 5 key.
D6	The 6 key.
D7	The 7 key.
D8	The 8 key.
D9	The 9 key.
Decimal	The . (DECIMAL) key.
Delete	The DEL (also known as DELETE) key.
Divide	The / (DIVIDE) key.
Down	The down arrow key.
E	The E key.
End	The END key.
Enter	The ENTER key.
Escape	The ESC (also known as ESCAPE) key.
F	The F key.
F1	The F1 key.
F10	The F10 key.
F11	The F11 key.
F12	The F12 key.
F2	The F2 key.
F3	The F3 key.
F4	The F4 key.
F5	The F5 key.
F6	The F6 key.

[values]	
F7	The F7 key.
F8	The F8 key.
F9	The F9 key.
G	The G key.
H	The H key.
Home	The HOME key.
I	The I key.
Insert	The INSERT key.
J	The J key.
K	The K key.
L	The L key.
Left	The left arrow key.
M	The M key.
Multiply	The * (MULTIPLY) key.
N	The N key.
None	A special value indicating no key.
NumPad0	The 0 key on the number pad.
NumPad1	The 1 key on the number pad.
NumPad2	The 2 key on the number pad.
NumPad3	The 3 key on the number pad.
NumPad4	The 4 key on the number pad.
NumPad5	The 5 key on the number pad.
NumPad6	The 6 key on the number pad.
NumPad7	The 7 key on the number pad.
NumPad8	The 8 key on the number pad.
NumPad9	The 9 key on the number pad.
O	The O key.
P	The P key.
PageDown	The PAGEDOWN key.
PageUp	The PAGEUP key.

[values]	
Q	The Q key.
R	The R key.
Right	The right arrow key.
S	The S key.
Shift	The SHIFT key.
Space	The SPACE key.
Subtract	The - (SUBTRACT) key.
T	The T key.
Tab	The TAB key.
U	The U key.
Unknown	A special value indicating the key is out of range of this enumeration.
Up	The up arrow key.
V	The V key.
W	The W key.
X	The X key.
Y	The Y key.
Z	The Z key.

KeyTimeSyntax

<table>
<tr><td>[values]</td><td></td></tr>
<tr><td><b>Uniform</b></td><td>Key frames will be uniformly spaced throughout the animation if they are all set to a key time of Uniform</td></tr>
<tr><td><b>Paced</b></td><td>Indicates that a key frame's duration should be calculated so as to keep a constant rate of change over the entire animation's duration.</td></tr>
<tr><td><b>[patterns]</b></td><td></td></tr>
<tr><td><b>(\d+\.)?\d\d?:\d\d?:((\d\d?)|(\d?\d?\.\d*))</b></td><td>The string may optionally begin with a decimal number and a dot. When present, this optional part indicates the number of days. The string always contains three parts separated by colons. The first two of these are one or two digit numbers specifying hours and minutes. The third part indicates the number of seconds and can be a two-digit integer, or a floating point number with 0, 1, or 2 digits before the point, and any</td></tr>
</table>

[values]	
	number of digits after the point. Examples: 0:0:1 One second 1.0:0:0 One day 0:0:0.5 Half a second 0:0:.5 Half a second 2.5:2:22 Two days, five hours, two minutes, 22 seconds
[is case sensitive]	True
\d+	A decimal number specifying the number of days. Examples: 1 One day 10 Ten days
[is case sensitive]	True
\+?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?\s*%	Decimal floating point number followed by a '%' symbol. (This is a percentage representing a proportion of the total duration of the animation, so it should have a value in the range from 0 to 100.)
[is case sensitive]	True

KeyTimeTypeSyntax

[values]	
TimeSpan	Each KeyTime is expressed as a TimeSpan value relative to the BeginTime of an animation sequence.
Uniform	The allotted total time for an animation sequence is divided evenly amongst each of the key frames.

LengthSyntax

[values]	
Auto	Indicates that automatic sizing should be used
[patterns]	
[+-]?((\d[\d,]*(\.\d*)?)|((\d[\d,]*)?\.\d+))([eE][+-]?\d+)? (\s*(PX|IN|CM|PT))?	A decimal floating point number, optionally followed by either 'in', 'cm', 'pt', or 'px'

LineStackingStrategySyntax

[values]	
BlockLineHeight	The stack height is determined by the block element line-height property value.
MaxHeight	The stack height is the smallest value that contains the extended block progression dimension of all the inline elements on that line when those elements are properly aligned. This is the default.

LogicalDirectionSyntax (4)

[values]	
Backward	Backward, or from right to left.
Forward	Forward, or from left to right.

LogSourceSyntax

[values]	
EndOfStream	The media reached the end of its stream.
MediaElementShutdown	The runtime shut down.
Pause	The media paused.
RequestLog	The RequestLog method was called.
RuntimeShutdown	The runtime shut down.
Seek	A seek operation occurred on the media; that is, playback was moved forward or backward.
SourceChanged	The source of the media changed.
Stop	The media stopped.

Matrix3DSyntax

[values]	
Identity	The identity matrix.
[patterns]	
(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){15}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	Sixteen decimal floating point values, separated by either a comma or whitespace.
[is case sensitive]	True

MatrixSyntax

[values]	
Identity	The identity matrix.
[patterns]	
(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){5}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	Six decimal floating point values, separated by either a comma or whitespace.
[is case sensitive]	True

MediaElementStateSyntax

[values]	
AcquiringLicense	The MediaElement is acquiring a license required to play DRM protected content. Once Uri) has been called, the MediaElement will remain in this state until Stream) has been called.
Buffering	The MediaElement is loading the media for playback. Its Position does not advance during this state. If the MediaElement was already playing video, it continues to display the last displayed frame.
Closed	The MediaElement contains no media. The MediaElement displays a transparent frame.
Individualizing	The MediaElement is in the process of ensuring that proper individualization components (only applicable when playing DRM protected content) are installed on the user's computer. See Digital Rights Management (DRM) for more information.
Opening	The MediaElement is validating and attempting to open the Uniform Resource Identifier (URI) specified by its Source property. While in this state, the MediaElement queues any Play, Pause, or Stop commands it receives and processes them if the media is successfully opened.
Paused	The MediaElement does not advance its Position. If the MediaElement was playing video, it continues to display the current frame.
Playing	The MediaElement is playing the media specified by its source property. Its Position advances forward.
Stopped	The MediaElement contains media but is not playing or paused. Its Position is 0 and does not advance. If the loaded media is video, the MediaElement displays the first frame.

MediaSampleAttributeKeysSyntax

[values]	
DRMInitializationVector	A Boolean value describing whether or not the given video frame is a keyframe.
DRMSubSampleMapping (4)	Describes the length of encrypted data and clear data (not encrypted data) in an H264 sample.
FrameHeight	Width of the video frame.
FrameWidth	Height of the video frame.
KeyFrameFlag	Data about the sample needed to decrypt it.

MediaSourceAttributesKeysSyntax

[values]	
CanSeek	A Boolean value that describes whether this source can seek.
DRMHeader	DRM data that the pipeline needs to initialize and decrypt correctly. This is the DRM header represented as a string.

[values]	
Duration	The length of playback time of this source as an integer in 100-nanosecond increments.

MediaStreamAttributeKeysSyntax

[values]	
CodecPrivateData	Codec data that the pipeline needs to initialize and render correctly. For video, this is other header information. For audio, this is the base16-encoded WaveFormatEx structure.
Height	The integer height a video frame is to be rendered at.
VideoFourCC	Data needed to instantiate a video codec. This is the four-character value also known as a FourCC.
Width	The integer width a video frame is to be rendered at.

MediaStreamSourceDiagnosticKindSyntax

[values]	
BufferLevelInBytes	Represents a download buffer in bytes.
BufferLevelInMilliseconds	Represents a download buffer in milliseconds.

MediaStreamTypeSyntax

[values]	
Audio	The stream is an audio stream.
Script	The stream is a script stream. Note: Currently script commands are not supported in MediaStreamSource.
Video	The stream is a video stream.

MessageBoxButtonSyntax

[values]	
OK	Displays only the OK button.
OKCancel	Displays both the OK and Cancel buttons.

MessageBoxResultSyntax

[values]	
Cancel	The user clicked the Cancel button or pressed ESC.
No	This value is not currently used.
None	This value is not currently used.

[values]	
OK	The user clicked the OK button.
Yes	This value is not currently used.

ModifierKeysSyntax

[values]	
Alt	The ALT key is pressed.
Apple	The Apple key (also known as the Open Apple key) is pressed.
Control	The CTRL key is pressed.
None	No modifiers are pressed.
Shift	The SHIFT key is pressed.
Windows	The Windows logo key is pressed.

OrientationSyntax

[values]	
Horizontal	The control or layout should be horizontally oriented.
Vertical	The control or layout should be vertically oriented.

PenLineCapSyntax

[values]	
Flat	A cap that does not extend past the last point of the line. Comparable to no line cap.
Round	A semicircle that has a diameter equal to the line thickness.
Square	A rectangle that has a height equal to the line thickness and a length equal to half the line thickness.
Triangle	An isosceles right triangle whose base length is equal to the thickness of the line.

PenLineJoinSyntax

[values]	
Bevel	Line joins use beveled vertices.
Miter	Line joins use regular angular vertices.
Round	Line joins use rounded vertices.

PixelFormatTypeSyntax (4)

[values]	
Format32bppArgb	The format uses 32 bits of color information per pixel and reports color information by using alpha, red, blue, and green channels.
Unknown	The format is unknown.

PlacementModeSyntax

[values]	
Bottom	Indicates that the preferred location of the tooltip is at the bottom of the target element.
Left	Indicates that the preferred location of the tooltip is at the left of the target element.
Mouse	Indicates that the preferred location of the tooltip is at the mouse pointer location.
Right	Indicates that the preferred location of the tooltip is at the right of the target element.
Top	Indicates that the preferred location of the tooltip is at the top of the target element.

PointCollectionSyntax

<table>
<tr><th>[patterns]</th><th></th></tr>
<tr><td>([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((((\s*,\s*)|\s+)([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)){2})*</td><td>Sequence of decimal floating point values, separated by either a comma or whitespace. The number of values in the sequence is a multiple of two.</td></tr>
<tr><td>[is case sensitive]</td><td>True</td></tr>
</table>

PointSyntax

<table>
<tr><th>[patterns]</th><th></th></tr>
<tr><td>([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)</td><td>Two decimal floating point values, separated by either a comma or whitespace.</td></tr>
<tr><td>[is case sensitive]</td><td>True</td></tr>
</table>

PropertyPathSyntax

[patterns]	
.*	Property paths have a syntax that involves balanced parentheses and balanced square brackets, and so it cannot be represented as a regular expression. The syntax takes the form of a list of property identifiers each separated by either a . or a /. Each property identifier is either an alphanumeric string of characters, or a sequence of characters enclosed in parentheses. Each property may optionally be followed by one or more square-bracket-enclosed alphanumeric strings. Examples: Property Property.More.Properties MyCollection[100] MyHashTable[foo].Bar Multi[10][20] Prop.Coll/Foo[10]

[patterns]	
[is case sensitive]	True

RectSyntax

<table>
<tr><td>[values]</td><td></td></tr>
<tr><td>**Empty**</td><td>Equivalent to a value of "0,0,0,0"</td></tr>
<tr><td>**[patterns]**</td><td></td></tr>
<tr><td>(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){2}((\+?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+))(\+?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)</td><td>Four decimal floating point values, separated by either a comma or whitespace. The last two digits must not be negative.</td></tr>
<tr><td>**[is case sensitive]**</td><td>True</td></tr>
</table>

RelativeSourceModeSyntax

[values]	
Self	Refers to the element on which you are setting the binding and allows you to bind one property of that element to another property on the same element.
TemplatedParent	Refers to the element to which the template (in which the data-bound element exists) is applied. This is similar to setting a TemplateBinding Markup Extension and is only applicable if the Binding is within a template.

RepeatBehaviorSyntax

<table>
<tr><td>[values]</td><td></td></tr>
<tr><td>**Forever**</td><td>Indicates that an animation should repeat indefinitely.</td></tr>
<tr><td>**[patterns]**</td><td></td></tr>
<tr><td>([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)x</td><td>A decimal floating point number followed by an 'x'. Indicates that the animation should repeat the specified number of times. Example: 2x Repeat twice 4.5x Repeat four and a half times</td></tr>
<tr><td>**[is case sensitive]**</td><td>True</td></tr>
<tr><td>(\d+\.)?\d\d?:\d\d?:((\d\d?)|(\d?\d?\.\d*))</td><td>The string may optionally begin with a decimal number and a dot. When present, this optional part indicates the number of days. The string</td></tr>
</table>

[values]	
	always contains three parts separated by colons. The first two of these are one or two digit numbers specifying hours and minutes. The third part indicates the number of seconds and can be a two-digit integer, or a floating point number with 0, 1, or 2 digits before the point, and any number of digits after the point. Examples: 0:0:1 Repeat for one second 1.0:0:0 Repeat for one day 0:0:0.5 Repeat for half a second 0:0:.5 Repeat for half a second 2.5:2:22 Repeat for two days, five hours, two minutes, 22 seconds
[is case sensitive]	True
\d+	A decimal number specifying the number of days. Examples: 1 Repeat for one day 10 Repeat for ten days
[is case sensitive]	True

RowOrColumnMajorSyntax

[values]	
ColumnMajor	Data in the table should be read column by column.
Indeterminate	The best way to present the data is indeterminate.
RowMajor	Data in the table should be read row by row.

SamplingModeSyntax

[values]	
Auto	The system selects the most appropriate sampling mode.
Bilinear	Use bilinear sampling.
NearestNeighbor	Use nearest neighbor sampling.

ScrollAmountSyntax

[values]	
LargeDecrement	Specifies that scrolling is performed in large decrements, which is equivalent to pressing the PAGE UP key or to clicking a blank part of a scrollbar. If the distance represented by the PAGE UP key is not a relevant amount for the control, or if no scrollbar exists, the value represents an amount equal to the size of the currently visible window.
LargeIncrement	Specifies that scrolling is performed in large increments, which is equivalent to pressing the PAGE DOWN key or to clicking a blank part of a scrollbar. If the distance represented by the PAGE DOWN key is not a relevant amount for the control, or if no scrollbar exists, the value represents an amount equal to the size of the currently visible region.

[values]	
NoAmount	Specifies that scrolling should not be performed.
SmallDecrement	Specifies that scrolling is performed in small decrements, which is equivalent to pressing an arrow key or to clicking the arrow button on a scrollbar.
SmallIncrement	Specifies that scrolling is performed in small increments, which equivalent to pressing an arrow key or to clicking the arrow button on a scrollbar.

ScrollBarVisibilitySyntax

[values]	
Auto	A ScrollBar appears and the dimension of the ScrollViewer is applied to the content when the viewport cannot display all of the content. For a horizontal ScrollBar, the width of the content is set to the ViewportWidth of the ScrollViewer. For a vertical ScrollBar, the height of the content is set to the ViewportHeight of the ScrollViewer.
Disabled	A ScrollBar does not appear even when the viewport cannot display all of the content. The dimension of the content is set to the corresponding dimension of the ScrollViewer parent. For a horizontal ScrollBar, the width of the content is set to the ViewportWidth of the ScrollViewer. For a vertical ScrollBar, the height of the content is set to the ViewportHeight of the ScrollViewer.
Hidden	A ScrollBar does not appear even when the viewport cannot display all of the content. The dimension of the ScrollViewer is not applied to the content.
Visible	A ScrollBar always appears. The dimension of the ScrollViewer is applied to the content. For a horizontal ScrollBar, the width of the content is set to the ViewportWidth of the ScrollViewer. For a vertical ScrollBar, the height of the content is set to the ViewportHeight of the ScrollViewer.

ScrollEventTypeSyntax

[values]	
EndScroll	Specifies that the Thumb was dragged to a new position and is now no longer being dragged by the user.
First	Specifies that the Thumb moved to the Minimum position of the ScrollBar.
LargeDecrement	Specifies that the Thumb moved a specified distance, as determined by the value of LargeChange. The Thumb moved to the left for a horizontal ScrollBar or upward for a vertical ScrollBar.
LargeIncrement	Specifies that the Thumb moved a specified distance, as determined by the value of LargeChange. The Thumb moved to the right for a horizontal ScrollBar or downward for a vertical ScrollBar.
Last	Specifies that the Thumb moved to the Minimum position of the ScrollBar.
SmallDecrement	Specifies that the Thumb moved a specified distance, as determined by the value of SmallChange. The Thumb moved to the left for a horizontal ScrollBar or upward for a vertical ScrollBar.
SmallIncrement	Specifies that the Thumb moved a specified distance, as determined by the value of SmallChange. The Thumb moved to the right for a horizontal ScrollBar or downward

[values]	
	for a vertical ScrollBar.
ThumbPosition	Specifies that the Thumb moved to a new position because the user selected Scroll Here in the shortcut menu of the ScrollBar.
ThumbTrack	The Thumb was dragged and caused a MouseMove event. A Scroll event of this ScrollEventType may occur more than one time when the Thumb is dragged in the ScrollBar.

SelectionModeSyntax

[values]	
Extended	The user can select multiple items by pressing a modifier key.
Multiple	The user can select multiple items without pressing a modifier key.
Single	The user can select only one item at a time.

SizeSyntax

[values]	
Empty	Equivalent to a value of "0,0"
[patterns]	
(\+?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)(\+?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)	Two decimal floating point values, separated by either a comma or whitespace.
[is case sensitive]	True

SolidColorBrushSyntax

StretchDirectionSyntax (4)	
[values]	
Both	The content stretches to fit the parent according to the Stretch property.
DownOnly	The content scales downward only when it is larger than the parent. If the content is smaller, no scaling upward is performed.
UpOnly	The content scales upward only when it is smaller than the parent. If the content is larger, no scaling downward is performed.

StretchSyntax

[values]	

[values]	
Fill	The content is resized to fill the destination dimensions. The aspect ratio is not preserved.
None	The content preserves its original size.
Uniform	The content is resized to fit in the destination dimensions while it preserves its native aspect ratio.
UniformToFill	The content is resized to fill the destination dimensions while it preserves its native aspect ratio. If the aspect ratio of the destination rectangle differs from the source, the source content is clipped to fit in the destination dimensions.

StringComparisonSyntax

[values]	
CurrentCulture	Compare strings using culture-sensitive sort rules and the current culture.
CurrentCultureIgnoreCase	Compare strings using culture-sensitive sort rules, the current culture, and ignoring the case of the strings being compared.
InvariantCulture	Compare strings using culture-sensitive sort rules and the invariant culture.
InvariantCultureIgnoreCase	Compare strings using culture-sensitive sort rules, the invariant culture, and ignoring the case of the strings being compared.
Ordinal	Compare strings using ordinal sort rules.
OrdinalIgnoreCase	Compare strings using ordinal sort rules and ignoring the case of the strings being compared.

StyleSimulationsSyntax

[values]	
BoldItalicSimulation	Bold and Italic style simulation.
BoldSimulation	Bold style simulation.
ItalicSimulation	Italic style simulation.
None	No font style simulation.

SupportedTextSelectionSyntax (4)

[values]	
Multiple	Supports multiple, disjoint text selections.
None	Does not support text selections.
Single	Supports a single, continuous text selection.

SweepDirectionSyntax

[values]	
Clockwise	Arcs are drawn in a clockwise (positive-angle) direction.
Counterclockwise	Aarcs are drawn in a counterclockwise (negative-angle) direction.

TabletDeviceTypeSyntax

[values]	
Mouse	Indicates the tablet device is a mouse.
Stylus	Indicates the tablet device is a stylus.
Touch	Indicates the tablet device is a touch screen.

TextAlignmentSyntax

[values]	
Center	Text is centered within the container.
Justify (4)	Text is justified within the container.
Left	Text is aligned to the left edge of the container.
Right	Text is aligned to the right edge of the container.

TextDecorationCollectionSyntax

[patterns]	
(NONE)|(OVERLINE|BASELINE|UNDERLINE|STRIKETHROUGH)?((\s*,\s*)(OVERLINE|BASELINE|UNDERLINE|STRIKETHROUGH))*	Either 'none', or a comma-separated list of any combination of overline, baseline, underline, and strikethrough.

TextHintingModeSyntax

[values]	
Animated	Turns off text rendering optimizations.

[values]	
Fixed	The default text render mode.

TextTrimmingSyntax (4)

[values]	
None	Text is not trimmed.
WordEllipsis	Text is trimmed at a word boundary. An ellipsis (...) is drawn in place of remaining text.

TextWrappingSyntax

[values]	
NoWrap	No line wrapping is performed.
Wrap	Line breaking occurs if a line of text overflows beyond the available width of its container. Line breaking occurs even if the standard line-breaking algorithm cannot determine any line break opportunity, such as when a line of text includes a long word that is constrained by a fixed-width container without scrolling.

ThicknessSyntax

<table>
<tr><th>[patterns]</th><th></th></tr>
<tr><td><b>(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+))?([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)</b></td><td>One decimal floating point value, or two values separated by either a comma or whitespace.</td></tr>
<tr><td><b>[is case sensitive]</b></td><td>True</td></tr>
<tr><td><b>(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){3}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)</b></td><td>Four decimal floating point values, separated by either a comma or whitespace.</td></tr>
<tr><td><b>[is case sensitive]</b></td><td>True</td></tr>
</table>

ToggleStateSyntax

[values]	
Indeterminate	The UI automation element is in an indeterminate state.
Off	The UI automation element is not selected, checked, marked, or otherwise activated.
On	The UI automation element is selected, checked, marked, or otherwise activated.

TouchActionSyntax

[values]	
Down	A touch action from a TOUCHEVENTF_DOWN message.

[values]	
Move	A touch action from a TOUCHEVENTF_MOVE message.
Up	A touch action from a TOUCHEVENTF_UP message.

TransformSyntax

<table>
<tr><td>[values]</td><td></td></tr>
<tr><td><b>Identity</b></td><td>The identity matrix.</td></tr>
<tr><td><b>[patterns]</b></td><td></td></tr>
<tr><td><b>(([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)((\s*,\s*)|\s+)){5}([+-]?((\d+(\.\d*)?)|(\d*\.\d+))([eE][+-]?\d+)?)</b></td><td>Six decimal floating point values, separated by either a comma or whitespace.</td></tr>
<tr><td><b>[is case sensitive]</b></td><td>True</td></tr>
</table>

UpdateSourceTriggerSyntax

[values]	
Default	The binding source is updated automatically when the binding target value changes.
Explicit	The binding source is updated only when you call the UpdateSource method.

ValidationErrorEventActionSyntax

[values]	
Added	A new ValidationError has occurred.
Removed	An existing ValidationError has been removed.

VerticalAlignmentSyntax

[values]	
Bottom	The element is aligned to the bottom of the parent's layout slot.
Center	The element is aligned to the center of the parent's layout slot.
Stretch	The element is stretched to fill the entire layout slot of the parent element.
Top	The element is aligned to the top of the parent's layout slot.

VideoOutputConnectorTypeSyntax (4)

[values]	
ComponentVideo	Component video connector.
CompositeVideo	Composite video connector.

[values]	
DisplayPortExternal	External DisplayPort connector.
DisplayPortInternal	Embedded DisplayPort connector.
Dvi	Digital video interface (DVI) connector.
Hdmi	High-definition multimedia interface (HDMI) connector.
Internal	Internal connector. The connection between the graphics adapter and the display device is permanent and not accessible to the user.
JapaneseDConnector	Japanese D connector (a connector conforming to the EIAJ RC-5237 standard).
Llvds	Low voltage differential signaling (LVDS) connector.
Other	An unknown connector type.
Sdi	SDI (serial digital image) connector.
SVideo	S-Video connector.
Tmds	
UdiExternal	External Unified Display Interface (UDI).
UdiInternal	Embedded Unified Display Interface (UDI).
Vga	Video Graphics Array (VGA) connector.

VirtualizationModeSyntax

[values]	
Recycling	Reuse the item containers.
Standard	Create and discard the item containers.

VisibilitySyntax

[values]	
Collapsed	Do not display the element, and do not reserve space for it in layout.
Visible	Display the element.

WaveFormatTypeSyntax (4)

[values]	
Pcm	Audio format uses pulse code modulation (PCM) encoding.

WindowInteractionStateSyntax

[values]	

[values]	
BlockedByModalWindow	The window is blocked by a modal window.
Closing	The window is closing.
NotResponding	The window is not responding.
ReadyForUserInteraction	The window is ready for user interaction.
Running	The window is running. This does not guarantee that the window is responding or ready for user interaction.

WindowResizeEdgeSyntax (4)

[values]	
Bottom	The bottom edge of the window.
BottomLeft	The bottom left corner of the window.
BottomRight	The bottom right corner of the window.
Left	The left edge of the window.
Right	The right edge of the window.
Top	The top edge of the window.
TopLeft	The top left corner of the window.
TopRight	The top right corner of the window.

WindowStartupLocationSyntax (4)

[values]	
CenterScreen	The application window is centered in the screen, and the Top and Left settings are ignored.
Manual	The application window is positioned according to the Top and Left settings.

WindowStateSyntax (4)

[values]	
Maximized	The application window is maximized to occupy the entire client area of the screen.
Minimized	The application window is minimized to the taskbar.
Normal	The application window is in its normal state, occupying screen space based on its Height and Width values.

WindowStyleSyntax (4)

[values]	

[values]	
BorderlessRoundCornersWindow	The window does not display a title bar or border, and the window corners are rounded.
None	The window does not display a title bar or border.
SingleBorderWindow	The window displays a title bar and border.

WindowVisualStateSyntax

[values]	
Maximized	Specifies that the window is maximized.
Minimized	Specifies that the window is minimized.
Normal	Specifies that the window is normal (restored).

References

[MS-XAML-2009], Microsoft Corporation, "XAML Object Mapping Specification 2009", April 2010

Index