

LESSON 2.1

10754 Microsoft .NET Fundamentals

Understand .NET Class Hierarchies

Lesson Overview

How are classes organized within the Microsoft® .NET Framework?

In this lesson, you will learn about:

- Class organization in the .NET Framework
- The *Object* class
- System classes

Guiding Questions

1. How are classes organized within the .NET Framework using assemblies and namespaces?
2. What kinds of functionality are provided by the System classes?

Anticipatory Set

With a partner, recall any projects or programs in which you added `using` statements to a Microsoft C# program or `Imports` statements to a Microsoft Visual Basic® program.

Assemblies

- An **assembly** is a collection of types and resources that are built to work together and form a logical unit of functionality.
- Each time you create a .NET application, you are creating an assembly.
 - There are other types of assemblies as well, such as class libraries.

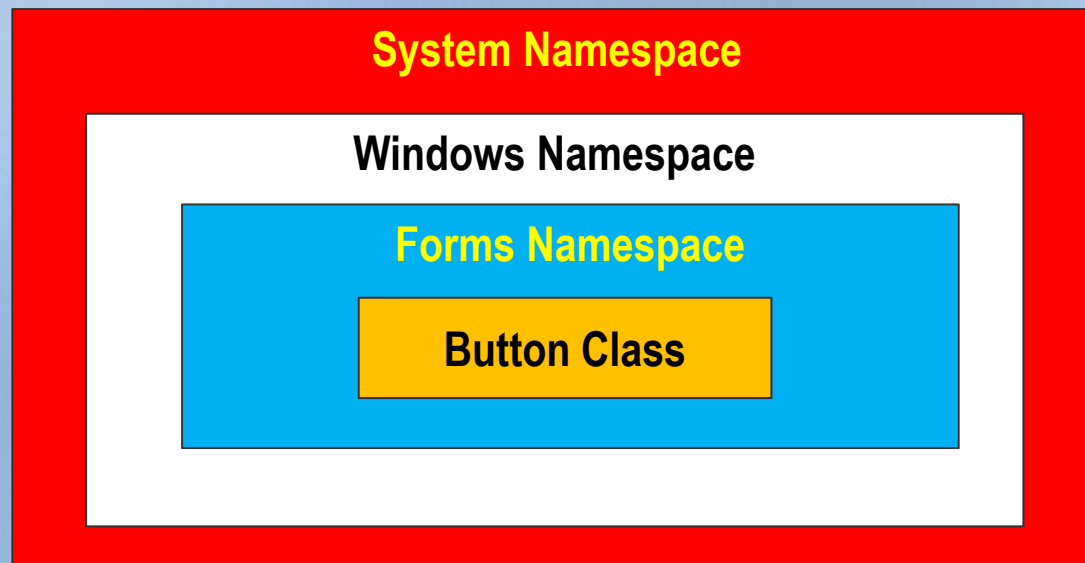
Namespaces

Namespaces are organizational units used for two purposes:

1. To organize the many classes within the .NET Framework
2. To help control scope and avoid name collisions (having two classes with the same name)

Namespace Example

- The `Button` type commonly used in Forms applications is nested within several namespaces:



- The fully qualified name for the `Button` class is `System.Windows.Forms.Button`.

Using and Imports Statements

- Because using fully qualified names can be cumbersome (as with the `Button` example), .NET languages provide a way to shorten references.
- In Microsoft C#, this is done by adding `using` statements at the top of your code.
- Microsoft Visual Basic uses the `Imports` statement.
- These statements allow developers to refer to `Button` rather than `System.Windows.Forms.Button`.

Object Class

- All classes in the .NET Framework are ultimately derived from a class called *Object*.
 - In other words, all classes in the .NET Framework—and all classes you define in your applications—inherit from *Object*, either directly or indirectly.
- The *Object* class includes a small number of methods that are often overridden but derived classes, such as `Equals (Object)`, `ToString ()`, and `Finalize ()`.

System Namespaces

- The *System* namespace contains fundamental classes and base classes that define commonly used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions.
- These classes and their members provide a foundation of classes, types, and methods for creating applications. This extensive library is one of the primary strengths of developing with the .NET Framework.

Assignment

- Complete the steps for creating an assembly, as described in the “Practice Creating an Assembly” section of <http://msdn.microsoft.com/en-us/library/ms973231.aspx>.
- Then, complete the “Practice Creating a Namespace” activity in the same article.
- When you finish, you will have a project named PowerLib.

Lesson Review

1. What are assemblies?
2. Why does the .NET Framework use namespaces?