UNIT 1
Assemblies, Namespaces and Class Libraries (metadata, DLL HELL)

Presented By : Aditya Joshi
Asst. Professor
Department of Computer Application
Graphic Era Deemed to be University

Dated : 23-01-2021

Assembly

- It’s a Deployment and configuration Unit in .NET.
- File Extension with .exe/.dll are known as assembly.

  exe(executables)

  .dll(dynamic link-library)
- Assembly includes meta data(information of source file, dependencies of that particular exe or dll).
- It Troubleshoots the DLL HELL Problem (not possible to store multiple versions of a particular file this is possible with the help of assembly)
- Assembly can be classified into two parts

1. Private Assembly
2. Shared Assembly

.NET assemblies contain

- The definition of types
- versioning information for the type
- meta-data and
- manifest.

Before we can use a class contained in an assembly in our application, we must add a reference to the assembly.

- mscorlib.dll
- System.dll
- System.Configuration.dll
- System.Web.dll
- System.Data.dll
- System.Web.Services.dll
- System.Xml.dll

Private Assembly

- Private assemblies is best suited or creating small applications
- It will use only a single version of assembly no management , no registration , versioning is not required
- Particular Single application will use single assembly, other applications are not affected.

Shared Assembly

- Many applications can use same assembly by sharing.
- It will not create any local copy of that assembly in each projects. So it reduce the disk and memory size.
- It is installed in GAC(Global Assembly Cache).
- It has a version number.
- It must have Strong unique name(this can be done by .snk(strong name key) file.

Assembly Structure

MSIL Code	Intermediate Language(IL) or Common Intermediate language (CIL)
Meta Data	Metadata in .Net is binary information which describes the characteristics of a resource . This information include Description of the Assembly , Data Types and members with their declarations and implementations, references to other types and members , Security permissions etc. A module's metadata contains everything that needed to interact with another module.
Manifest	During the compile time Metadata created with Microsoft Intermediate Language (MSIL) and stored in a file called a Manifest. Manifest is a file that containing Metadata about .NET Assemblies. Assembly Manifest contains a collection of data that describes how the elements in the assembly relate to each other.
Resources	A resource is any nonexecutable data that is logically deployed with an app.

Namespace

- Namespaces is a logical group of related classes that can be used by any other language targeting the Microsoft .Net framework . It is more used for logical organization of your classes. Namespaces are a way of grouping type names and reducing the chance of name collisions.

Hierarchy and Fully-Qualified Names

The fully qualified name of a class is constructed by concatenating the names of all the namespaces that contain the type. For e.g.

the fully qualified name of the TextBox class is System.Windows.Forms.TextBox .

That means TextBox class is contained in the Forms namespace that is contained in the Windows namespace that is contained in the root System namespace.

The C# using Keyword

- The using keyword states that the program is using the names in the given namespace. For eg

we are using System Namespace in our programs. The class Console is defined there we just write

```
Console.WriteLine(“Hello”);
```

by including the namespace at the top of the code

Like

```
using System;
```

Note: Namespace are placed within assemblies

.NET Framework Class Library

The Framework Class Library or FCL provides the system functionality in the **.NET Framework** as it has various classes, data types, interfaces, etc. to perform multiple functions and build different types of applications such as desktop applications, web applications, mobile applications, etc. The Framework Class Library is integrated with the Common Language Runtime (CLR) of the .NET framework and is used by all the .NET languages such as **C#**, F#, Visual Basic .NET, etc.

Categories in the Framework Class Library

The functionality of the Framework Class Library can be broadly divided into **three** categories

i.e

1. *utility features written in .NET*,
2. *wrappers around the OS functionality* and
3. *frameworks*.

- **Utility Features:** The utility features in the FCL includes various collection classes such as list, stack, queue, dictionary, etc. and also classes for more varied manipulations such as Regex class for regular expressions.
- **Wrappers Around OS functionality:** Some of the features in the FCL are wrappers around the underlying Windows OS functionality. These include the classes for using the file system, the classes to handle the network features, the classes to handle I/O for console applications, etc.
- **Frameworks:** There are various frameworks available in the FCL to develop certain applications. For example, ASP.NET is used to develop web applications, Windows Presentation Foundation (WPF) is used to render user interfaces in Windows applications and so on.

Namespaces in the Framework Class Library

Namespaces in the Framework Class Library are a group of related classes and interfaces that can be used by all the *.NET framework* languages. Some of the namespaces in the FCL along with their description is given as follows:

NAMESPACE	DESCRIPTION
Accessibility	The Accessibility namespace is a part of the managed wrapper for the COM accessibility interface.
Microsoft.Activities	The Microsoft.Activities namespace provides support for Windows Workflow Foundation applications.
Microsoft.CSharp	The Microsoft.CSharp namespace has support for compilation and code generation for the C# source code.
Microsoft.JScript	The Microsoft.JScript namespace has support for compilation and code generation for the JScript source code.
Microsoft.VisualBasic	The Microsoft.VisualBasic namespace has support for compilation and code generation for the VisualBasic source code.
System	The System namespace has base classes for definition of interfaces, data types, events, event handlers, attributes, processing exceptions etc.
System.Activities	The System.Activities namespace handles the creation and working with activities in the Window Workflow Foundation using various classes.
System.Collections	The System.Collections namespace has multiple standard, specialized, and generic collection objects that are defined using various types.

System.Configuration	The System.Configuration namespace handles configuration data using various types. This may include data in machine or application configuration files.
System.Data	The System.Data namespace accesses and manages data from various sources using different classes.
System.Drawing	The System.Drawing namespace handles GDI+ basic graphics functionality. Various child namespaces also handle vector graphics functionality, advanced imaging functionality, etc.
System.Globalization	The System.Globalization namespace handles language, country, calendars used, format patterns for dates, etc. using various classes.
System.IO	The System.IO namespaces support IO like data read/write into streams, data compression, communicate using named pipes etc. using various types.
System.Linq	The System.Linq namespace supports Language-Integrated Query (LINQ) using various types.
System.Media	The System.Media namespace handles sound files and accessing the sounds provided by the system using various classes.

System.Net	The System.Net namespace provides an interface for network protocols, cache policies for web resources, composing and sending e-mail etc. using various classes.
System.Reflection	The System.Reflection namespace gives a managed view of loaded methods, types, fields, etc. It can also create and invoke types dynamically.
System.Security	The System.Security namespace has the .NET security system and permissions. Child namespaces provide authentication, crytographic services etc.
System.Threading	The System.Threading namespace allows multithreaded programming using various types.
XamlGeneratedNamespace	The XamlGeneratedNamespace has compiler-generated types that are not used directly from the code.