Net FrameWork Part 3

13. What is mean by Namespace?

Namespace Logically group classes, it avoids name clashes between classes.

Example : most of the general purpose .net base classes are in a namespace called System. The base class Array is in this namespace is accessed with full name System.Array.

14. What is difference between Assembly and Namespace?

Assembly is physical grouping of classes. Namespace logically groups classes.

Single assembly can have different namespaces

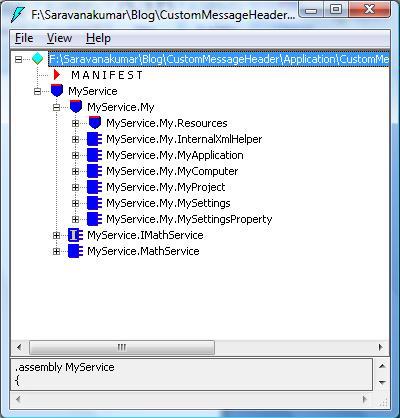
Sample namespace can be used in different assembly. E.g the assembly mscorlib and system contain the namespace System.Threading

15. What is the difference between an executable assembly and a class library?

An executable assembly exists as the .exe file while a class library exists as the .dll file. Executable assembly represent executable applications having some entry (e.g., Main() method in C#). A class library, on the other hand, contains components and libraries to be used inside various applications. A Class library cannot be executed and thus it does not have any entry point.

16. What is ILDASM?

ILDASM(Intermediate Language DisAssembler ), this is inbuild tool to view content and manifest of the assembly. We can run the ILDASM by running following exe "C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\NETFX 4.0 Tools\ildasm.exe"



ILDASM

Note : Currently I am using V4, file path will vary for different version of SDK installation.

17. What is mean by Manifest?

Manifest is used to describe the information about the assembly, it contains following information.

Assembly name - Aids in versioning and visibility scope.

Version information - The version number is integrated into the assembly's identity.

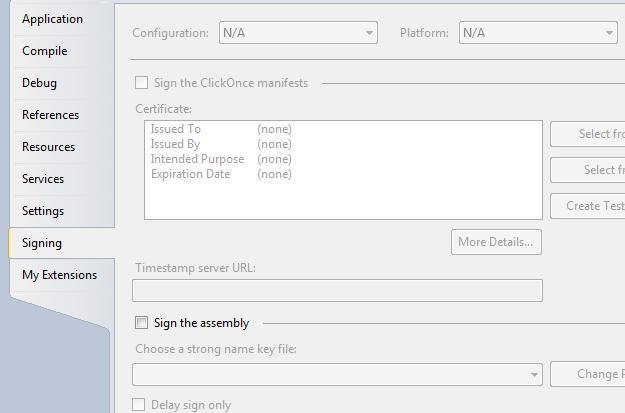
Types - Boundaries and scopes of methods, classes, properties, events and attributes.

Locale - Information describing language/culture.

Reference - provides information for type references in an assembly and other referenced assemblies.

Cryptographic Hash - Public key encoded hash acting as version/security check.

Security Permissions - The permissions within the assembly determine the permissions that can be granted for all aspects of the assembly contents.



18. How will you created shared assembly?

Shared assembly can be created by signing the assembly. Sets to created shared assembly

Create new class library project using visual studio

Navigate to the property page of the class library

Select "Signing" tab

Shared Assmebly

Select "Sign the assembly" check-box

Now select < New >... from "Choose a strong name key file" dropdown

Enter new Signing key file name and click Ok

Shared Assmebly

Next the build the project. Now the shared assembly is ready to use in different project.

19. What is the use of Shared Assembly?

If you want to use the same assembly in different projects, we can create a shared assembly and placed inside the GAC(Global assembly Catch). So that assembly is access by all the application. Private assembly also be used in different projects, but we need to copy the private assembly files to different application folder. But if we are using Shared assembly, the assembly file remains in single location.

Shared assembly is highly secured, only administrator can uninstall the shared assembly.

20. What is GAC?

GAC(Global assembly catch) is used to store .Net assembly. It is located in "C:\Windows\assembly"

Assembly located in GAC is shared by multiple applications

Adding an Assembly to GAC

"gacutil -i (assembly\_name)", where (assembly\_name) is the DLL name of the project.

21. What is mean by Delay signing?

During development process, usually private key will not be exposed to the developer due to security reason. In this kind of scenario, we will go with delay signing.Delay signing allows you to place a shared assembly in the GAC by signing the assembly with just the public key. This allows the assembly to be signed with the private key at a later stage, when the development process is complete and the component or assembly is ready to be deployed. This process enables developers to work with shared assemblies as if they were strongly named, and it secures the private key of the signature from being accessed at different stages of development.

E.g

VB.Net(Assemblyinfo.vb)

< Assembly: AssemblyKeyFileAttribute("myKey.snk") >

< Assembly: AssemblyDelaySignAttribute(True) >

C#(Assemblyinfo.cs)

[assembly: AssemblyKeyFileAttribute("myKey.snk")]

[assembly: AssemblyDelaySignAttribute(true)]