

1 :: Define .Net Assembly?

It is a primary unit of deployment in a Microsoft .NET Framework application. It is called as building block of an application which provides all required execution information to common language runtime.

An assembly perform following functions:

It contains IL code that gets executed by common language runtime.

It forms a security boundary.

An assembly is the unit at which permissions are requested and granted.

It ensures type safety by establishing name scope for types at the runtime.

It contains version information.

It allows side-by-side execution of multiple versions of same assembly.

Assemblies can be static or dynamic.

Static assemblies are created when the program is compiled using .Net compiler. It exists as PE file either in .exe or .dll. However, dynamic assemblies are created at runtime and run from the memory without getting saved on the disk.

2 :: What does the dot net assembly contain?

An assembly contains following information:

Assembly manifest: Information about the assembly.

Type metadata: Information about the types.

IL Code

Resource files.

An assembly manifest contains the following information:

Identity of the assembly

Types and resources

Files

Security permissions

3 :: Define private assembly and a shared dot net assembly?

A private assembly is stored in the application's directory and used by a single application. A share assembly can be used by multiple applications and is stored in the Global assembly cache, a repository of assemblies maintained by the .Net Framework.

4 :: What are Satellite Assemblies?

Satellite assemblies provide an application the multilingual support. Satellite assemblies contain alternate sets of resources to be used in the application for different cultures.

5 :: What do you understand by side-by-site execution of assembly?

This means multiple version of same assembly to run on the same computer. This feature enables to deploy multiple versions of the component.

6 :: How do you create a resource-only assembly?

Resources are nonexecutable data in an application and the data can be updated without recompiling application. Resource assemblies can be created as follows:

Add resource files to an empty project.

Build the project.

The resource will get compiled into assembly.

7 :: Explain how to retrieve resources using ResourceManager class?

ResourceManager class is used to retrieve resources at run time.

Create a ResourceManager with resource file name and the resource assembly as parameters.

After having created, you can use ResourceManager.GetString method to retrieve a string.

Use the ResourceManager.GetObject method to retrieve images and objects from a resource file.

8 :: Define Strong Name. How do you apply a strong name to assembly?

A strong name means generating public key in order to provide unique name to the assembly.

The name is used to provide global name to the assembly and allows it to be shared amongst several different applications.

The key generated include assembly's name, the version number, the developer's identity, and a hash number.

The developer's identity identifies the author of the assembly.

The hash checks if the assembly is tempered since it is created.

The key pair that defines the strong name is created using the Strong Name utility, Sn.exe.

To sign an assembly with a strong name

Create Key pair using the Strong Name utility, Sn.exe.

Open the AssemblyInfo file of your project.

Use the AssemblyKeyFileAttribute to specify the path to the key file for your project.

Build your assembly. The strong name will be generated and signed to the assembly.

9 :: Define Global Assembly Cache in dot net?

Global Assembly Cache is the place holder for shared assembly. If an assembly is installed to the Global Assembly Cache, the assembly can be accessed by multiple

applications. In order to install an assembly to the GAC, the assembly must have to be signed with strong name.

10 :: How do you install assembly to the Global Assembly Cache?

Followings are the steps to install assembly to the GAC.

Sign assembly with a strong name using strong name utility, sn.exe.

Open the AssemblyInfo file for your project.

Use the AssemblyKeyFileAttribute to specify the path to the key file for your project.

Build your assembly. Install the assembly to GAC by using gacutil utility e.g. gacutil -i abc.dll

www.downloadmela.com