What is An Assembly

An Assembly is a basic building block of .Net Framework applications. It is basically a compiled code that can be executed by the CLR. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An Assembly can be a DLL or exe depending upon the project that we choose.

Assemblies are basically the following two types:

1. Private Assembly
2. Shared Assembly

**1. Private Assembly**

It is an assembly that is being used by a single application only. Suppose we have a project in which we refer to a DLL so when we build that project that DLL will be copied to the bin folder of our project. That DLL becomes a private assembly within our project. Generally, the DLLs that are meant for a specific project are private assemblies.

**2. Shared Assembly**

Assemblies that can be used in more than one project are known to be a shared assembly. Shared assemblies are generally installed in the GAC. Assemblies that are installed in the GAC are made available to all the .Net applications on that machine.

However, there are two more types of assemblies in .Net, Satellite Assembly, and Shared Assembly.

**GAC**

GAC stands for Global Assembly Cache. It is a memory that is used to store the assemblies that are meant to be used by various applications.

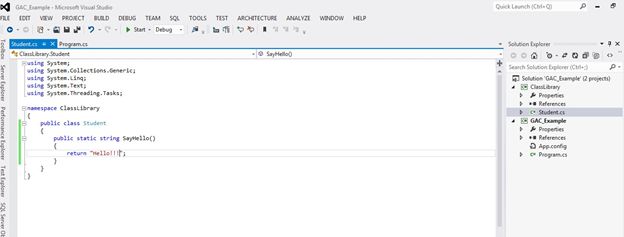
Every computer that has CLR installed must have a GAC. GAC is a location that can be seen at “C:\Windows\assembly” for .Net applications with frameworks up to 3.5. For higher frameworks like 4 and 4.5 the GAC can be seen at:

“C:\Windows\Microsoft.NET\assembly\GAC\_MSIL”.

**Installing an assembly into the GAC**

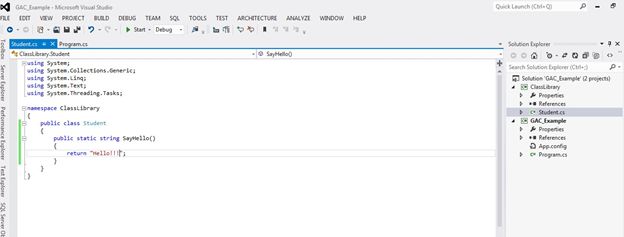
For installing an assembly into the GAC we first need to generate strong names for the assembly because only a strong name assembly is installed in the GAC. All other assemblies are known as weak named assemblies and they cannot be stored in the GAC.

For generating a strong name assembly we first create a console application type project. In that solution, we will add one more project (Class Library).



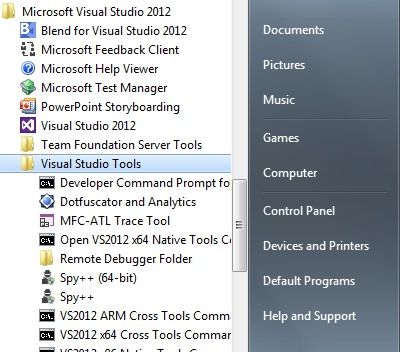
In the Class Library project, I created a class Student that contains a method that returns a string. Now I will add a reference of the Class Library to my Console application project and will use this method.

If we see the bin folder of our Console Application project then we will see that the ClassLibrary DLL is present there (It will come when we build our project). This DLL is a Private Assembly now.

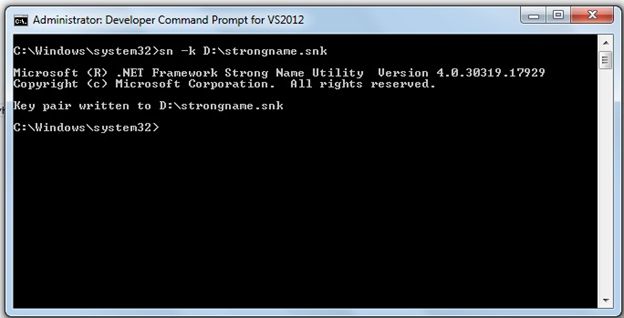


**Generating Strong Name**

To make our assembly strongly named we need to generate a unique key-value pair, for that we need to open the Visual Studio command prompt. We can open that from the Start button by going to "All programs" -> "Visual Studio 2012" (or any version that you have)  -> "Visual Studio Tools" -> "Developer Command Prompt for Visual Studio" and run that as an administrator.



Then type the command sn –k “Path where you want that key to be generated”



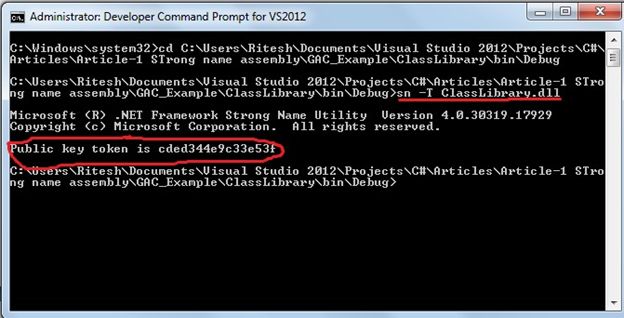
This will generate the unique key pair that we will use in our application.

Now move to the Visual Studio Class Library Project and open the AssemblyInfo.cs file that is residing in the Properties folder of your project. In the bottom of that file add a new attribute that refers to the key pair file that we have just created.

[assembly: AssemblyKeyFile("D:\\strongname.snk")]

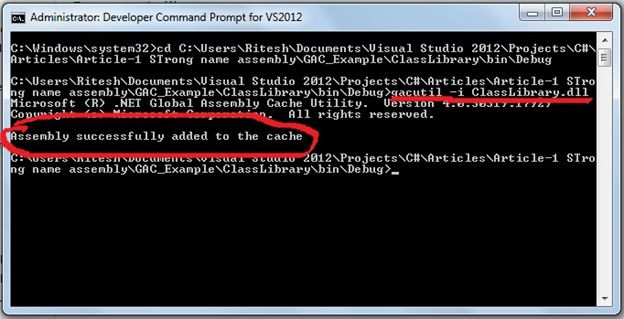


Now build your Class Library project and ensure it builds successfully. If it builds then it means that your ClassLibrary DLL has become strongly typed. If you want to check that then open the Visual Studio command prompt, copy the path of the "bin.Debug" folder of your ClassLibrary project. Now paste the path into the command prompt with the cd command (refer to the following screenshot). Now type the command sn –T “ClassLibrary.dll”. After typing this command you will get a public key token that indicates that your assembly is Strongly Named and it is ready to install in the GAC.



Now to install the assembly in the cache type the following command in the command prompt:

gacutil –i "Name of the Class Library"



Now the Assembly is successfully installed on the system. If you want to see that you can see that in the location “C:\Windows\Microsoft.NET\assembly\GAC\_MSIL”.

**Note**: If you are using the framework greater than 4.0 then you can search the preceding given path but if you are using a framework lesser than 4.0 then your assembly will be installed at “C:\Windows\assembly”.

