Programming in C#
With Visual Studio 2010 Vol 1

Lesson 01

Assembly

- Compiled C# -> MSIL (Microsoft Intermediate Language) that is readable by the CLR.
- All .NET compilers produce assemblies
- Collection of types and resources that work together and form a logical unit of functionality
- Can be Executable or a Library

Assembly Versioning

- Major
- Minor
- Build
- Revision

Assembly Signing

- Methods
    - o Sign Tool
    - o Assembly signing functionality in VS2010
- Protects assemblies from modification
- Signed Assemblies con be included in the signed assembly in the Global Assembly Cache (GAC) so you can share the assembly with multiple applications.
- Guarantees the assembly name is unique
-

CLR

1. Class Loader – locates and loads all assemblies that the application requires. (MSIL)
2. MSIL-to-Native compiler verifies the MSIL codes and then compiles all assemblies into machine code ready for execution.
3. Code Manager – loads the executable assembly and runs the Main method.
4. Garbage Collector – provides automatic lifetime memory management of all objects that your application creates. The GC disposes of any objects that your application is no longer using.

5. Exception Handler - provides structure exception handling for .NET applications, which is integrated with Windows structured exception handling.

.NET Framework Tools

Code Access Security Policy Tool (Caspol.exe)

Enables users to modify the machine, user, and enterprise security policy. This can include defining a custom permission set and adding assemblies to the full trust list.

Certificate Creation Tool (Makecert.exe)

Enables users to create x.509 certificates for use in their development environment. Typically, you can use these certificates to sign your assemblies and define Secure Sockets Layer (SSL) connections.

Global Assembly Cache Tool (Gacutil.exe)

Enables users to manipulate the assemblies in the GAC. This can include installing and uninstalling assemblies in the GAC so that multiple applications can access them.

Native Image Generator (Ngen.exe)

Enables users to improve the performance of .NET applications. The Native Image Generator improves performance by precompiling assemblies into images that contain processor-specific machine code. The CLR can then run the precompiled images instead of using just-in-time (JIT) compilation. Alternatively, if you use JIT compilation, your code is compiled just before it is executed.

MSIL Disassembler (Ildasm.exe)

Enables users to manipulate assemblies, such as determining whether an assembly is managed, or disassembling an assembly to view the compiled MSIL code.

Strong Name Tool (Sn.exe)

Enables users to sign assemblies with strong names.The Strong Name Tool includes commands to create a new key pair, extract a public key from a key pair, and verify assemblies.

Lesson 02