Microsoft .Net

The visual studio version 6.0 has several drawbacks especially in developing web applications. To overcome these drawbacks of visual studio 6.0, Microsoft started a project with the name **Next Generation Windows Services (NGWS)** in 1998. But later it was renamed to **Visual Studio.Net** in 2000 and first beta version of .net was released in 2000. In the year 2002 the complete version of .net was released and was named as **visual studio.net 2002**. In this version of .net there is no support for mobile and device application development. In the year 2003 Microsoft releases next version of .net and was named as **visual studio.net 2003** and it has the support for mobile and device applications. Again in the year 2005 Microsoft releases the next version of .net named as **visual studio.net 2005** and it have several new features when compared to .net 2003. Microsoft is planning to release the next version of .net in February 2008 with the name **visual studio.net 2008**.

To work with .net i.e. for both creating and executing .net applications, **.net framework** is needed. All the components required for both developing and executing .net applications are in **.net framework**. **.net framework** also has different versions. .net 2002 comes with **.net framework version 1.0**, .net 2003 comes with **.net framework 1.1** and .net 2005 comes with **.net framework 2.0**. In **Windows Vista** operating system, **.net framework 3.0** was provided. The next version of .net **visual studio.net 2008** comes with **.net framework 3.5**.

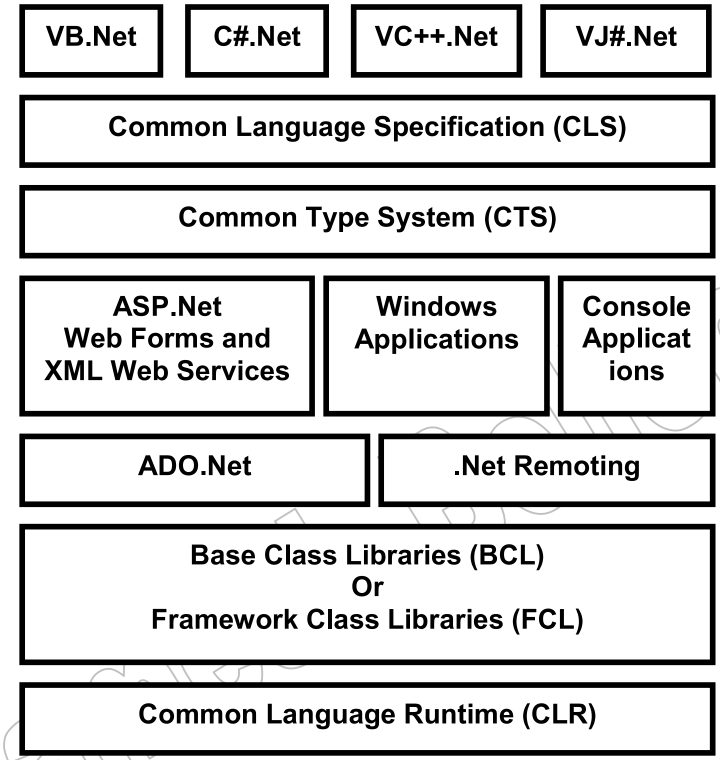
On the operating systems like **Windows 98, Windows me, Windows NT, Unix, Linux and Solaris** .net framework can be installed but not visual studio.net. On these operating systems if you want to develop .net applications then you have to use text editor like notepad. For the operating systems like **Unix, Linux** and **Solaris**, .net framework called **Mono** is used and it is developed by **Novell** company but not Microsoft.

On the operating systems like **Windows XP with Service Pack 2, Windows 2000 with Service Pack 4, Windows 2003 with Service Pack 1** and **Windows Vista** operating systems you can install both .net framework 2.0 as well as visual studio.net 2005. Windows 2003 operating system has built in .net framework version 2.0 and for vista operating system the built in .net framework version is 3.0. Hence while installing .net 2005 on vista, you have to install 2 service packs for .net 2005 on vista provided by Microsoft for compatibility between .net framework 2.0 and .net framework 3.0.

Visual Stuio.net 2008 that comes with **.Net Framework 3.5** can be installed on operating systems like **Windows XP with SP2, Windows 2003 With SP1, Windows Vista** and **Windows 2008 server** that will be released in February 2008 along with visual studio.net 2008.

.Net Framework

.net framework is an integrated component of operating system that supports development and execution of next generation windows services and XML web services. .net framework architecture is as follows.


VB.Net
C#.Net
VC++.Net
VJ#.Net
Common Language Specification (CLS)
Common Type System (CTS)
ASP.Net Web Forms and XML Web Services
Windows Applications
Console Applicat ions
ADO.Net
.Net Remoting
Base Class Libraries (BCL) Or Framework Class Libraries (FCL)
Common Language Runtime (CLR)

Common Language Specification (CLS)

CLS provides a **set of rules** and an **object oriented programming model** to be followed by every .net language. When a language follows these set of rules and object oriented programming model then that language is called as **CLS Compliant** i.e. nothing but .net language. To create classes and perform inheritance in a .net language, CLS is required. Because it provides the object oriented programming model.

Common Type System (CTS)

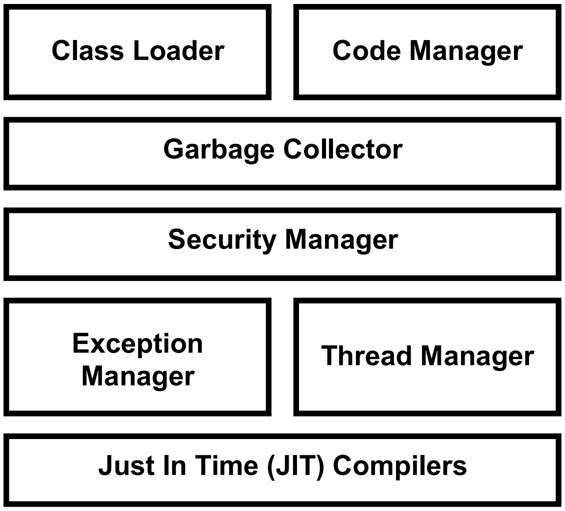
Every data type in .net is represented either in the form a class or structure. These classes and structures provide the members that can perform various operations on that particular data type. For example string class for string data type provides the members like **ToUpper, ToLower** to convert the string to uppercase and lowercase. All the classes and structures related to data types in .net are available in CTS. Without CTS it is not possible to declare variables in any .net language. Even every .net language provides its own keywords for data types, they all use same classes and structures available in CTS.

Base Class Libraries (BCL)

In C and C++ header files are available that provides library functions that can be directly used in C or C++ program. Similar to header files in C and C++, in .net framework we have BCL that provide classes, structures, enumerations, interfaces and delegates that can be directly used in a .net application. A class library is divided in to namespaces and namespaces will contain classes, structures, enumerations, interfaces and delegates. When you want to use a type from class library in your application, then you have to add the reference of namespace that contains the type you want to use to your application and for this vb.net provides **imports** statement and C# provides **using** statement. This is similar to writing **#include** statement in C and C++. .net framework provides nearly 15000 to 20000 classes and without using at least one class from these it is not possible to create a .net application.

Common Language Runtime (CLR)

CLR is the runtime environment in which .net applications are executed. CLR forms as a layer on top of the operating system and provides its services between operating system and .net. CLR Architecture is as follows.



Class Loader

Class loader is responsible to verify what are the class libraries referred by .net application and load all those class libraries in to memory to make them available for .net application. Class loader is not responsible for loading your application code in to memory.

Code Manager

Code manager is responsible for loading your .net application code in to memory and manage it until .net application was closed. The .net application code that is loaded in to memory and is running under the control of code manager is called as **managed code** and the .net application code that is loaded in to memory but is not under the control of code manager is called as **unmanaged code**. Examples of unmanaged code are **visual studio 6.0 (COM) component used .net** and **pointers in C#**.

Garbage Collector

Garbage collector is responsible for automatic memory management for the .net application. Allocating memory for the variable and objects created in .net application and de-allocate memory when their life time is expired is the responsibility of garbage collector.

Security Manager

You can provide access to files and folders created in your application based on user id and password provided while logging in to windows operating system and this security related code is executed by security manager.

Exception Manager

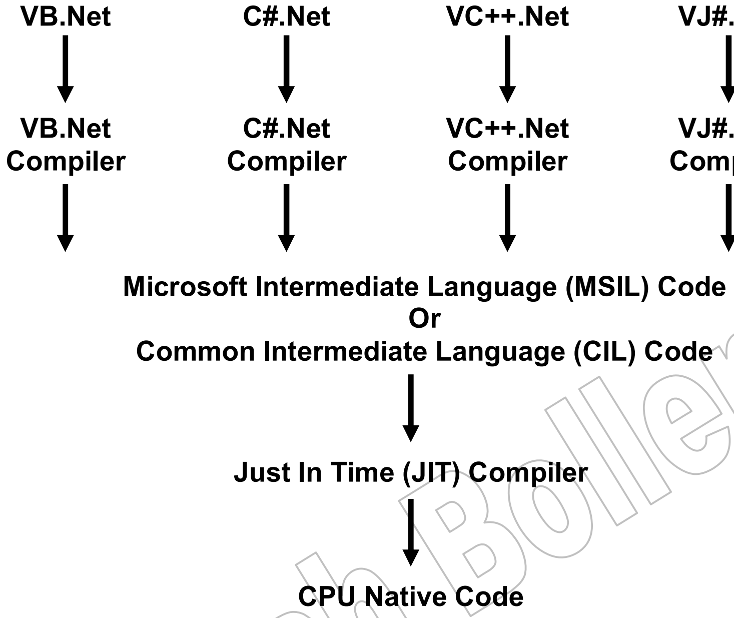
Exception manager is responsible for Exception handling in .net. it is responsible for verifying whether there is any possibility for exceptions, execute exception handling code if available and terminate the program abnormally if exception handling code is not available.

Thread Manager

When you want to perform multiple tasks at a time in you applications, then you need to create multiple threads in the application and in this case managing and executing those multiple threads is the responsibility of thread manager.

Just In Time (JIT) Compiler

Every .net language has its own compiler and it compiles application according to the syntax of that language. But all .net language compilers will produce same code after compilation i.e. **Microsoft Intermediate Language(MSIL) code.** Advantage of MSIL code is it is CPU platform independent and can be executed on any CPU platform. But MSIL code is not understandable to processor. This MSIL code is compiled by **JIT Compiler** to **CPU native code** understandable by processor and then the application is executed. CLR contains multiple JIT compilers, one for each CPU platform.


VB.Net
C#.Net
VC++.Net
VJ#.Net
VB.Net Compiler
C#.Net Compiler
VC++.Net Compiler
VJ#.Net Compiler
Microsoft Intermediate Language (MSIL) Code
Or
Common Intermediate Language (CIL) Code
Just In Time (JIT) Compiler
CPU Native Code