

MICROSOFT

The .net framework 2.0

Multi-Dimensional Solution Platform



The .NET Framework is a software development platform by Microsoft. It's based on the use of a virtual machine that runs an IL (Intermediate Language). Programs are assembled to produce IL which is further distributed to users to run on virtual machines. C#, VB.NET and C++ compilers are accessible from Microsoft for developing IL.

.NET has given new functionalities and tools to the application programming interface. These improvisations facilitate the process of developing applications for Windows, as well as for web services. .NET framework provides a contemporary, insightful, object-oriented API. High flexibility standards have been implemented in its design to make it multi-language compatible.

.NET Framework 2.0

Microsoft released the latest version, .Net 2.0 Framework on October 27, 2005. It is not only a redesign of the technology but also a model of Microsoft's .NET agenda. The earlier .NET version was based on a Rapid Application Development model. The primary goal for developing .NET 2.0 Framework was to allow developers to write 70% less code as compared to what was required using .NET v1.x.

.NET 2.0 Development Environment:

Visual Studio 2005 is an integrated development environment for .NET 2.0. Similar to Visual Studio 2005, an open source integrated development environment also exists, known as SharpDevelop. Some of its features provide better enhancement capabilities as compared to Visual Studio 2005.

.NET v1.x was mainly considered a Rapid Application Development system; whereas the new.NET 2.0 version acts as a base for the new generation Microsoft platform: WinFX.

SYSTEM REQUIREMENTS:

Supporting Operating Systems:

Windows 2000 Service Pack 3
Windows 98
Windows 98 Edition II
Windows ME
Windows Server 2003
Windows XP Service Pack 2

Required Software:

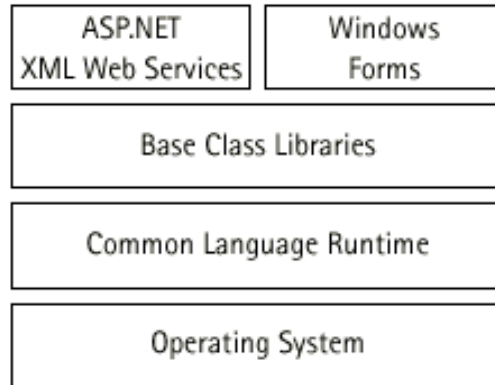
- Windows Installer 3.0, except for Windows 98/ME, which require Windows Installer 2.0. Windows Installer 3.1.
- Microsoft Internet Explorer 5.01 for all installations of the .NET Framework.

Disk Space Requirements:

- 280 MB (x86), 610 MB (x64)

Enterprise Infrastructure:

.NET provides a platform-independent structure for software development, with numerous built-in features including Internet integration and security enhancement features. It completely depends on software components and the component based programming models.



The .Net Framework

The two principal components of .NET are the Common Language Infrastructure (CLI) and the Common Language Runtime (CLR). The CLI is a set of terms for a runtime environment, which includes a common type system, Common Intermediate Language (CIL) code, which is machine independent and a base class library.

The CLR offers a platform for managing code that conforms to the CLI. The CLR translates the CIL into a local machine code before its execution.

CIL is self-describing through .NET metadata. The metadata is checked by CLR to make sure that the correct method is being deployed. Developers are enabled to generate metadata; though it is also generated by the language compilers in some cases.

If a language implementation generates CIL, it can be hosted using the CLR and can then interact with data produced by any other .NET language.

.Net Remoting & XML WebServices:

The .NET remoting infrastructure is a conceptual approach towards internal communication process. Microsoft .NET remoting offers a rich and extensible framework for objects residing in other application domains, in different processes, and in different machines for seamless communication amongst one another. .NET remoting offers a robust and a simple programming model and runtime support for providing a transparency to these interactions.

XML Web service is the elementary structure required for the process of distributed computing on the Internet. Open standards and focused attention on communication and association between people and applications have created an environment where XML Web services are being deployed as the platform for integrating applications. Multiple synchronized XML Web services from diverse sources are deployed to develop applications.

.NET Security:

.NET security mechanism has two basic features:

- **Code access security:**

Code access security is evidence oriented, which is linked with a particular assembly. Characteristically the evidence is the source of the assembly. It is either directly installed on the native machine or can be downloaded from the internet. Evidence is used by code access security to verify the permissions granted to the code. Other code can demand that calling code is granted a specified permission. The demand results in the CLR performing a call stack walk. In this process every assembly of each method in the call stack is verified for the required permission.

- **Validation & Verification:**

Validation & Verification are the two tests performed by CLR when the assembly is loaded. During the validation process the CLR verifies the presence of metadata & CIL and also verifies for the validity of internal tables. By deploying a few algorithms the verification mechanism checks to see if the code does any thing that is 'unsafe'.

.NET Languages:

The CLI has been designed in a way that it supports any object-oriented programming language by sharing a common object model and a large common class library. .NET 2.0 supports more than 40 programming languages.

Some of the languages built-in the .Net Framework are:

- C# 2.0
- JScript .NET
- J#
- Managed C++

- Visual Basic .NET

.NET 2.0 Framework also supports a few third party languages, such as:

- A#.
- APL
- Boo
- COBOL
- Component Pascal
- Delphi 8 and Delphi 2005
- Eiffel
- F#
- Forth
- FORTRAN
- IKVM, Java
- IronPython, Python
- Lexico
- Lisp
- Mercury
- Mondrian
- Nemerle
- Oberon
- Perl
- PHP
- RPG
- Smalltalk