Introduction to .NET Framework

It is a software technology.

It is an application development & execution Environment.

It is a Microsoft managed code development environment.

It provides set of base classes and language compilers which are necessary for programming in .net environment.

In .NET Framework, the set of base classes is called an Assembly.

So that, .NET Framework is a collection of assemblies.

Why .NET Framework?

It is a software technology, which is used for development of various software applications.

Console application

Windows Forms

Web forms

Mobile forms

Enterprise applications [client/server]

It is a single environment for various applications with multiple technologies.

It also enables us to build connected systems with various devices.

How .NET Framework

[Application Development process in .NET]

.NET Languages

These provide the syntaxes, which are used to implement the source code with .net base classes.

C#,VB,F# ,VC++,PERL,VJ#,COBOL,C,C++ and etc.

.NET 4.5 has 70+ languages support.

It enables us to build one application with multiple languages.

It has cross language compatibility.

ADO.Net

(ActiveX Data Object.Net)

It is a database connectivity model.

It enables us to establish connection to various database engines and perform SQL actions.

ASP.NET

[Active Server pages.NET]

It is a server side web technology.

It is a web programming model.

It uses one of the .net compatible languages for server script

Distributed Computing Technologies

Remoting

Web Services

COM+

MSMQ

WCF [It is a unification of all .net distributed technologies.]

These all are used to implement client/server applications [Connected Systems].

UI: User Interface

Presentation Layer

Here, we perform Input and Output.

BL: Business logic Layer

It consists of functionality.

Whenever we perform something on UI, these actions will takes place.

DAL: Data Access Layer

It consists of database connections and database operations.

Features of .NET Framework

- It is a platform Independent.
- It is a language independent.
- It is a user friendly environment.
- It is highly performable.
- It is highly securable.
- It has fully objected oriented programming support.
- It has xml support.
- It has disconnected database connectivity model.[ADO.Net]
- It enables us to develop simple console applications to standard enterprise applications.
- It allows easy development of web application with rich user interface(Asp.net)
- It enables us to build background tasks [Windows services]
- It enables us to build web callable functionalities [Web services]
- It enables us to build globalized applications [Globalization and Localization]
- It supports multithreaded programming.
- It supports interoperability.
- It supports multi-device Applications.

Execution of .Net application

Source code =>language compiler or Visual Studio MSIL => CLR =>JIT compiler =>native code.

demo.cs => csc demo.dll/demo.exe => [MSIL + metadata] => CLR => JIT compiler => machine code

Note: -

.Net is a platform independent because of MSIL [Microsoft Intermediate language]

Note: -

In .Net framework, we can develop applications with any of .net languages.

We can also develop one application with multiple languages.

Note:-

.Net F/W provides set of readymade classes, which are ready to use in our application.

Array

Math

File

StreamReader

Socket

ADO.NET

ADO.NET is a disconnected database connectivity model.

It enables us to maintain data in the local system in the form of XML document without any connection to database engine.

Later, we can establish connection to server, and we can read from xml and store in Db permanently.

Asp.Net

It is used to dynamic and data driven web application.

It also enables us to develop rich user interface with inbuilt controls in asp.net.

Calendar

AdRotator

Themes [share common styles]

Master pages [share common content]

Grid Controls

Windows Services

These are automatic services.

These can be started automatically at time of kernel startup.

These are also called background task.

These don't any user interface.

Data base server

E-mail server

Login server

Web server, Mailing server and etc...

Web services

These are used to implement, web methods.

These can be accessed by any user throughout the world.

These are used to build connected systems (heterogeneous systems]

Globalization

It enables us to build to application with multiple cultures.

It automatically set the culture based on the location.

The default culture: en-us (US English)

Multithreaded program

In .net environment, we can run two or more threads simultaneously.

It is supported by .net CLR Thread support Manager.

Interoperability

It provides the communication between .net framework and COM Components.

e.g.

CCW (COM Callable Wrapper)

RCW (Runtime Callable Wrapper)

CCW:

We can expose .NET components to COM Application.

RCW:

We can consume COM components in .NET Application.

Multi Device support

.Net enables us to build an application, which can be hosted in various devices

Mobiles [smart Devices]

Pagers

PDA

[This is done through .NET Compac Framework]

Fully Object Oriented

In .Net Framework, we do programming using any .net language.

Here, we implement business logic with class & objects.

It supports all the Object Oriented programing features like: Encapsulation, Abstract, Inheritance, Polymorphism and etc…

In .Net, no application exists without class.

Components of .NET Framework

1) BCL OR FCL

2) CLR

BCL

Base class Library

It consists of set of inbuilt classes.

We use these base classes and implement the source code.

Here, the collection of base classes is called an assembly.

Note: -

Assembly is a physical collection of logical units.

It is a physical or runtime file (.dll or .exe).

The logical units are called namespaces.

The namespace is a logical grouping of types.

The Type is a representation of data.

In .Net, we have 5 types:

Class, interface, enum, struct, delegate

Assembly

Namespaces

Types [classes]

Services [methods]

mscorlib.dll

System

Console

WriteLine()

Note: -

In .Net, an assembly can be either .exe /.dll

In .Net, an assembly can be single file or multi file assembly.

An Assembly can be private or public:

Private Assembly

It is applicable for single application.

It builds and packaged into a single application.

Public or Shared Assembly

It is applicable for multiple applications.

Only one copy of public assembly exists in a machine, but used in multiple applications.

.DLL

Dynamic link library

It is a library file.

It is a reusable file.

It is not executable.

.exe

It is an executable file.

It is executed and I/O is performed.

It is not reusable.

It requires entry point.

Sample Assemblies

mscorlib.dll

System.Core.dll

System.Data.dll

System.Data.OracleClient.dll

System.Xml.dll

System.Runtime.Remoting.dll

System.Web.dll

System.ServiceProcess.dll

Sample Namespaces

mscorlib.dll

System

System.Collections

System.Text

System.Collections.Generic

System.IO

System.Threading

..

mscorlib.dll

- System
  - Object
  - Array
  - Math
  - Console
  - Boolean
  - Byte
  - String
  - Int16

mscorlib.dll

- System
  - Console
    - WriteLine()
    - ReadLine()
    - Read()
    - Write()

CLR Common language runtime

CLR is responsible for managing and executing codes written in .NET languages, and forms the basis of .NET architecture.

The CLR is responsible for object activation, performing security checks on objects at run time, memory management and garbage collection.

CLR is the runtime engine that loads the required classes, performs just-in-time compilation, and performs security checks and other runtime activities.

C#: csc

Vb: vbc

J# :jsc

F# : fsc

CLR is the common execution engine for the .net framework.

It is the main part of .NET F/W.

Components of CLR

Class Loader

It will allocate memory to the members during runtime.

Type checker

It will check the type compatibility while integrating multiple source codes.

Debug Manager

It will take care of application execution process.

Garbage collector

In .Net, if there is free object continuously for 5 minutes, it automatically destroyed.

Exception Manager

It provides Exception handlers.

By using Exception handlers, we can avoid abnormal termination.

Thread support Manager

It enables us to execute two or more threads simultaneously.

It allows multithreading and multitasking.

Security Manager

It will take care of type safety and code access security.

COM Marshaller

It enables us to perform interoperability.

JIT Compiler

Just In time Compiler

It interprets the MSIL code into the machine code.

Assemblies (.exe/.dll)

It consists of MSIL + metadata

This combination is called PE (portable executable).

This is located at Manifest.

To View this MSIL and Metadata, we use ILDASM Utility command.

ILDASM library.dll/.exe

MSIL [IL or CIL]

Compiler generated code from high level language

Platform independent

Used by CLR

Gets compiled to platform dependent executable

It is the Managed Code.

Note:-

COM components are Unmanaged Code.

Metadata

It shows the descriptions and properties of an assembly.

Information about program structure is language-agnostic, so that it can be referenced between languages and tools, making it easy to work with code written in a language you are not using.

Compiler generate metadata

Metadata contains description of all types contained in an assembly

Metadata describes all classes and class members that are defined in the assembly, and the classes and class members that the current assembly will call from another assembly.

Manifest

- Assembly manifest contains:
  - Assembly name
  - Version number
  - Security identity etc.
  - Every assembly contains a collection of data that describes how the elements in the assembly relate to each other.
  - The assembly manifest contains this assembly metadata.
  - An assembly manifest contains all the metadata needed to specify the assembly's version requirements and security identity, and all metadata needed to define the scope of the assembly and resolve references to resources and classes.

CTS Common Type System

It provides set of System types, which are common to any .net compatible language.

A set of types and operations that are shared by many programming languages.

System.Byte

System.Int16

System.Int32

System.Int64

System.Single

System.Double

System.String

System.Object

System.DateTime

System.Char

CLS Common language Specification

It provides certain language specifications, which are common to any .net language.

A set of base rules which any language targeting the CLI should conform to in order to interoperate with other CLS-compliant languages.

e.g. Keywords, signatures, primitive types, interface, constructor invocation, overloading, properties and etc…

Software's for .NET F/W

1).NET Framework SDK

It provides set of base classes and languages compilers.

We write the source in any text editor and compile it with .NET Framework SDK command prompt.

.NET F/W SDK 4.5 is the current Version

2) Microsoft Visual Studio

It is a RAD Tool

It is an IDE for .NET f/w.

.NET F/W SDK is installed automatically along with Visual Studio

It provides the following features

1) It provides Drag and Drop feature.

2) It provides Excellent Debugging

3) It provides Intellisence feature

4) It provides indentation.

5) It provides Integration with source controls like VSS and TFS .

6) It allows easy deployment.

7) It allows easy project management.

8) It's current version is: Visual Studio 2012

3).NET Redistributable file

It consists of set of runtime files.

It enables us to execute the .net application.

It is intended to run application.

It is given to the client along with the product.