DOTNET TECHNOLOGIES

UNIT – 1

Introduction:

.NET is a framework to develop software applications. It is designed and developed by Microsoft and the first beta version released in 2000.

It is used to develop applications for web, Windows, phone. Moreover, it provides a broad range of functionalities and support.

This framework contains a large number of class libraries known as Framework Class Library (FCL). The software programs written in .NET are executed in the execution environment, which is called CLR (Common Language Runtime). These are the core and essential parts of the .NET framework.

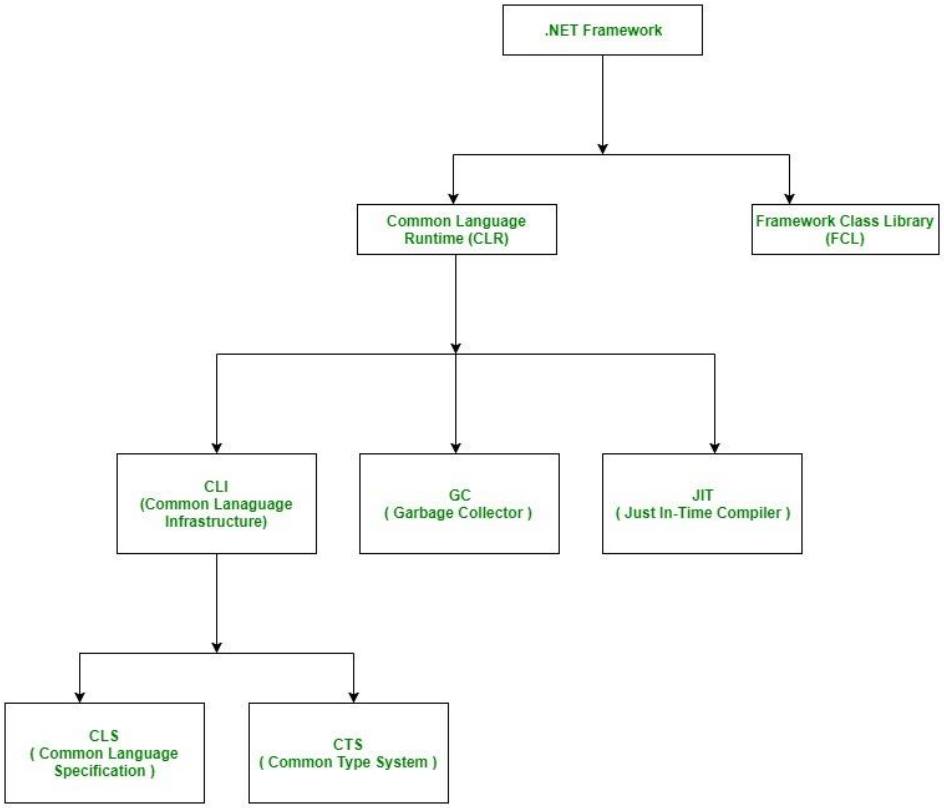
This framework provides various services like memory management, networking, security, memory management, and type-safety.

The .NET Framework supports more than 60 programming languages such as C#, F#, VB.NET, J#, VC++, JScript.NET, APL, COBOL, Perl, Oberon, ML, Pascal, Eiffel, Smalltalk, Python, Cobra, ADA, etc.

- Following is the .NET framework Stack that shows the modules and components of the Framework.

The .NET Framework is composed of four main components:

1. Common Language Runtime (CLR)
2. Framework Class Library (FCL),
3. Core Languages (WinForms, ASP.NET, and ADO.NET), and
4. Other Modules (WCF, WPF, WF, Card Space, LINQ, Entity Framework, Parallel LINQ, Task Parallel Library, etc.)



Features of .NET:

- It is a platform neutral framework.
- It is a layer between the operating system and the programming language.
- It supports many programming languages, including VB.NET, C# etc.
- .NET provides a common set of class libraries, which can be accessed from any .NET based programming language. There will not be separate set of classes and libraries for each language. If you know any one .NET language, you can write code in any .NET language.
- In future versions of Windows, .NET will be freely distributed as part of operating system and users will never have to install .NET separately.

Microsoft Intermediate Language (MSIL):

- The Microsoft Intermediate Language (MSIL) also goes by the name Common Intermediate Language (CIL).
- It is typically a set of platform-independent instructions created from source code by a language-specific compiler.

- It is produced by a variety of compilers (C#, VB,.NET, and so on).
- The ILDasm (Intermediate Language Disassembler) software included in the.NET Framework SDK (FrameworkSDKBinildasm.exe) allows users to view MSIL code in a human-readable format.
- We may view MSIL code in any.NET executable file (EXE or DLL) using this application.

MSIL is machine agnostic and may be turned into native code quickly. A just-in-time (JIT) compiler is included in the CLR, which turns MSIL code into native machine code. As a result, MSIL must be translated by a JIT compiler before being executed on the CPU. For each supported machine architecture, a JIT compiler is available. Any supported machine will run the same MSIL.

Framework Base Classes:

The .NET Framework has an extensive set of class libraries. This includes classes for:

1. **Data Access:** High Performance data access classes for connecting to SQL Server or any other OLEDB provider.
2. **XML Supports:** Next generation XML support that goes far beyond the functionality of MSXML.
3. **Directory Services:** Support for accessing Active Directory/LDPA using ADSI.
4. **Regular Expression:** Support for above and beyond that found in Perl 5.
5. **Queuing Supports:** Provides a clean object-oriented set of classes for working with MSMQ.

These class libraries use the CLR base class libraries for common functionality.

Base Class Libraries:

The Base class library in the .NET Framework is huge. It covers areas such as:

1. **Collection:** The System.Collections namespaces provides numerous collection classes.
2. **Thread Support:** The System.Threading namespace provides support for creating fast, efficient, multi-threaded appliction.

3. **Code Generation:** The System.CodeDOM namespace provides classes for generating source files in numerous language. ASP.NET uses these classes when converting ASP.NET pages into classes, which are subsequently compiled.
4. **IO:** The System.IO provides extensive support for working with files and all other stream types.
5. **Reflection:** The System.Reflection namespace provides support for load assemblies, examining the type with in assemblies, creating instances of types, etc.
6. **Security:** The System.Security namespace provides support for services such as authentication, authorization, permission sets, policies, and cryptography.

These base services are used by application development technologies like ASP.NET to build their security infrastructure.

Meta Data:

Metadata is data about the data or documentation about the information which is required by the users. In data warehousing, metadata is one of the essential aspects.

Metadata is used for building, maintaining, managing, and using the data warehouses. Metadata allow users access to help understand the content and find data.

Metadata in .Net is binary information which describes the characteristics of a resource. This information includes Description of the Assembly, Data Types and members with their declarations and implementations, references to other types and members, Security permissions etc.

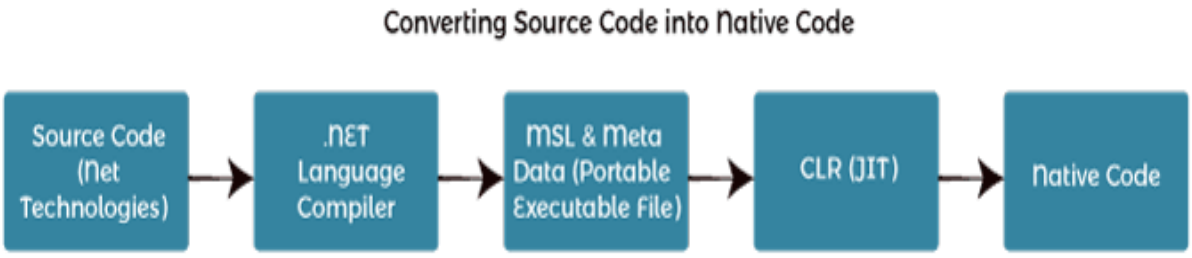
Meta data in execution time:

1. During the compile time Metadata created with Microsoft Intermediate Language (MSIL) and stored in a file called a Manifest.
2. Both Metadata and Microsoft Intermediate Language (MSIL) together wrapped in a Portable Executable (PE) file.

3. During the runtime of a program Just In Time (JIT) compiler of the Common Language Runtime (CLR) uses the Metadata and converts Microsoft Intermediate Language (MSIL) into native code.
4. When code is executed, the runtime loads metadata into memory and references it to discover information about your code's classes, members, inheritance, and so on.
5. Metadata eliminating the need for Interface Definition Language (IDL) files, header files, or any external method of component reference.

CLR (Common Language Runtime):

1. It is a program execution engine that loads and executes the program. It converts the program into native code. It acts as an interface between the framework and operating system. It does exception handling, memory management, and garbage collection. Moreover, it provides security, type-safety, interoperability, and portablility.
2. The CLR is the heart of .NET framework. It is .NET equivalent of Java Virtual Machine (JVM). It is the runtime that converts a MSIL (Micro Soft Intermediate Language) code into the host machine language code, which is then executed appropriately.



The CLR provides a number of services that include:

1. Loading and execution of codes
2. Memory isolation for application
3. Verification of type safety
4. Compilation of IL into native executable code
5. Providing metadata Automatic garbage collection

6. Enforcement of Security
7. Interoperability with other systems
8. Managing exceptions and errors
9. Provide support for debugging and profiling

.NET Namespaces:

Namespaces are the way to organize .NET Framework Class Library into a logical grouping according to their functionality, usability as well as category they should belong to, or we can say Namespaces are logical grouping of types for the purpose of identification.

- The .NET Framework Class Library (FCL) is a large collection of thousands of Classes.
- These Classes are organized in a hierarchical tree.
- The System Namespaces is the root for types in the .NET Framework.
- We can uniquely identify any Class in the .NET Framework Class Library (FCL) by using the full Namespaces of the class.
- In .NET languages every program is created with a default Namespaces. Programmers can also create their own Namespaces in .NET languages.

Common Type System (CTS):

The Common Type System (CTS) is a standard for defining and using data types in the .NET framework. CTS defines a collection of data types, which are used and managed by the run time to facilitate cross-language integration.

CTS provides the types in the .NET Framework with which .NET applications, components and controls are built in different programming languages so information is shared easily.

In contrast to low-level languages like C and C++ where classes/structs have to be used for defining types often used (like date or time), CTS provides a rich hierarchy of such types without the need for any inclusion of header files or libraries in the code.

CTS is a specification created by Microsoft and included in the European Computer Manufacturer‘s Association standard. It also forms the standard for implementing the .NET framework.

CTS supported data types:

- System.Byte
- System.Int16
- System.Int32
- System.Int64
- System.Single
- System.Double
- System.Object
- System.Char
- System.String
- System.Decimal
- System.Boolean

Common Language Specification (CLS):

CLS is a part of the specifications of the .NET Framework. CLS was designed to support language constructs commonly used by developers and to produce verifiable code, which allows all CLS-compliant languages to ensure the type safety of code. CLS includes features common to many object-oriented programming languages.

The Common Language Specification (CLS) is a fundamental set of language features supported by the Common Language Runtime (CLR) of the .NET Framework.

It forms a subset of the functionality of common type system (CTS) and has more rules than defined in CTS.

CLS represents the guidelines to the compiler of a language, which targets the .NET Framework.

CLS-compliant code is the code exposed and expressed in CLS form. Even though various .NET languages differ in their syntactic rules, their compilers generate the Common Intermediate Language instructions, which are executed by CLR.

CLS allows flexibility in using non-compliant types in the internal implementation of components with CLS-compliant requirements.

.NET Applications:

1. Console apps.
2. Windows GUI apps (Windows Forms).
3. Windows Presentation Foundation (WPF) apps.
4. ASP.NET apps.
5. Windows services.
6. Service-oriented apps using Windows Communication Foundation (WCF).
7. Workflow-enabled apps using Windows Workflow Foundation (WF)