# Introduction t o .NET Framework

**.NET**is a software framework which is designed and developed by **Microsoft**. The first version of .Net framework was 1.0 which came in the year 2002. In easy words, it is a virtual machine for **compiling and executing programs** written in **different languages** like [C#](https://www.geeksforgeeks.org/introduction-to-c/), VB.Net etc.  
It is used to **develop Form-based/desktop/windows applications, Web-based applications, and Web services**. There is a variety of programming languages available on the .Net platform, VB.Net and [C#](https://www.geeksforgeeks.org/introduction-to-c/) being the most common ones are . It is used to build applications for Windows, phone, web etc. It provides a lot of functionalities and also supports industry standards.

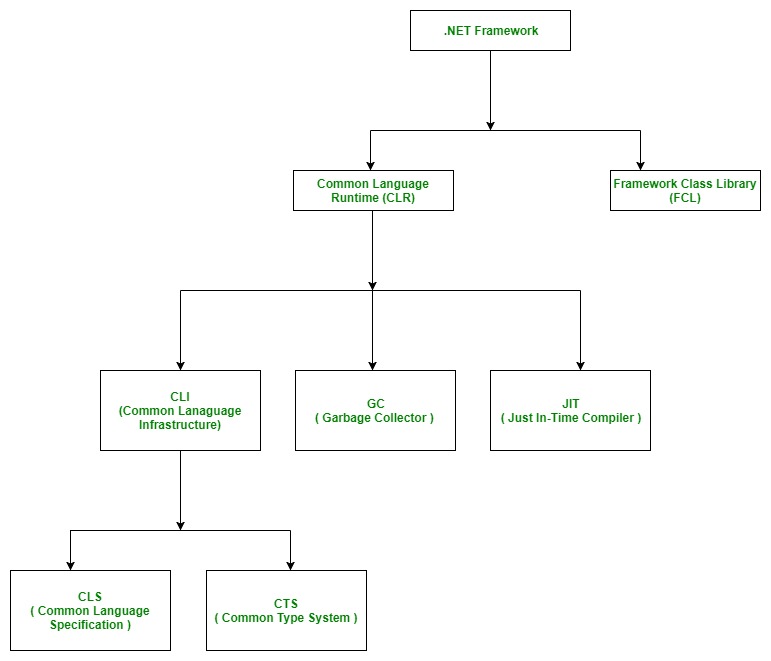
.NET Framework supports more than 60 programming languages in which 11 programming languages are designed and developed by Microsoft. The remaining [Non-Microsoft Languages](https://bitbucket.org/brianritchie/wiki/wiki/.NET%20Languages) which are supported by .NET Framework but not designed and developed by Microsoft.

11 Programming Languages which are designed and developed by Microsoft are:

* CSharp.NET c#
* VB.NET
* C++.NET
* J#.NET
* F#.NET
* JSCRIPT.NET
* WINDOWS POWERSHELL
* IRON RUBY
* IRON PYTHON
* C OMEGA
* ASML(Abstract State Machine Language)

COMMON LANGUAGE RUNTIME (CLR) IN C#

* CLR is the basic and Virtual Machine component of the .NET Framework. It is the run-time enviornment in the .NET Framework that runs the codes and helps in making the development process easier by providing the various services. Basically, it is responsible for managing the execution of .NET programs regardless of any .NET programming language. Internally, CLR implements the VES(Virtual Execution System) which is
* defined in the Microsoft’s implementation of the CLI(Common Language Infrastructure).
* The code that runs under the Common Language Runtime is termed as the Managed Code. In other words, you can say that CLR provides a managed execution enviornment for the .NET programs by improving the security, including the cross language integration and a rich set of class libraries etc. CLR is present in every .NET framework verison. Below table illustrate the CLR version in .NET framework.



CLR VERSIONS .NET FRAMEWORK VERSIONS

1.0 1.0

1.1 1.1

2.0 2.0

2.0 3.0

2.0 3.5

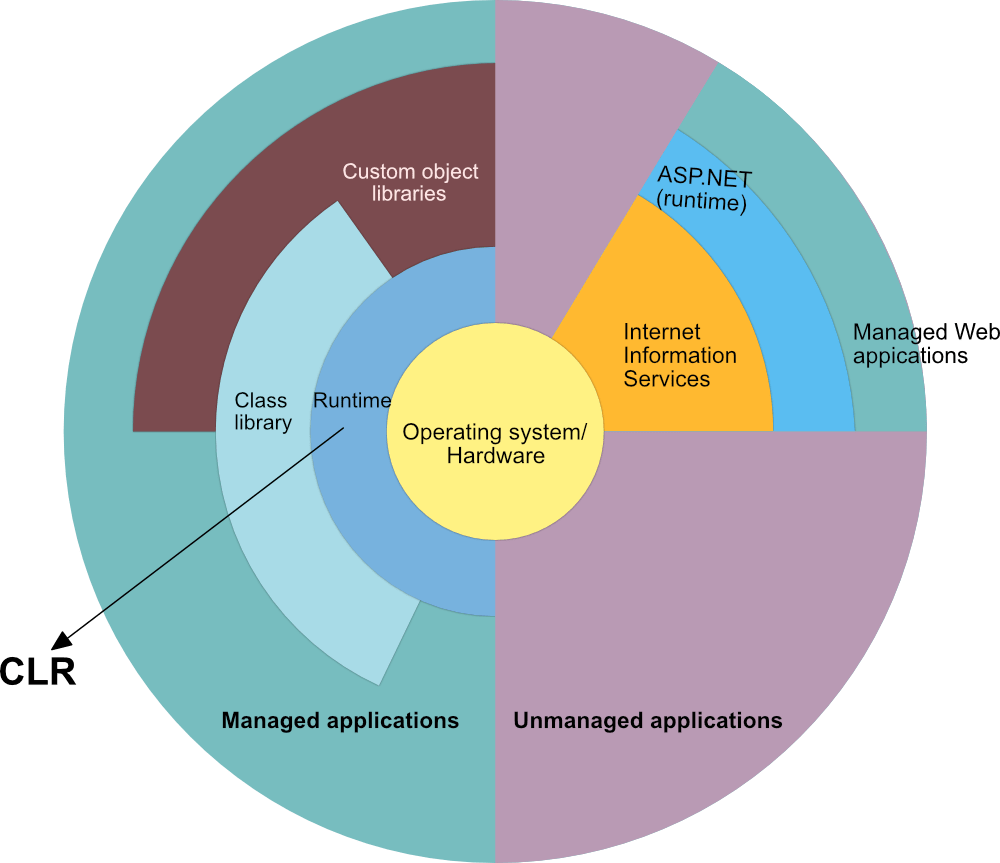
4 4

4 4.5(also 4.5.1 & 4.5.2)

4 4.6(also 4.6.1 & 4.6.2)

4 4.7(also 4.7.1 & 4.7.2)

Below diagram illustrate that how CLR is associated with the operating system/hardware along with the class libraries. Here, runtime is actually CLR.



ROLE OF CLR IN THE EXECUTION OF A C# PROGRAM

* Suppose you have written a C# program and save it in a file which is known as the Source Code.
* Language specific compiler compiles the source code into the MSIL(Microsoft Intermediate Language) which is also know as the CIL(Common Intermediate Language) or IL(Intermediate Language) along with its metadata. Metadata includes the all the types, actual implementation of each function of the program. MSIL is machine independent code.
* Now CLR comes into existence. CLR provides the services and runtime enviornment to the MSIL code. Internally CLR includes the JIT(Just-In-Time) compiler which converts the MSIL code to machine code which further executed by CPU. CLR also uses the .NET Framework class libraries. Metadata provides the information about the programming language, enviornment, version, and class libraries to the CLR by which CLR handles the MSIL code. As CLR is common so it allows an instance of a class that written in a different languague to call a method of the class which written in another langauge.

MAIN COMPONENTS OF CLR

As the word specify Common which means CLR provides a common runtime or execution environment as there are more than 60 .NET programming languages.

MAIN COMPONENETS OF CLR:

* COMMON LANGUAGE SPECIFICATION (CLS)
* COMMON TYPE SYSTEM (CTS)
* GARBAGE COLLECTION (GC)
* JUST IN – TIME COMPILER (JIT)

Common Language Specification (CLS):

* It is responsible for converting the different .NET programming language syntactical rules and regulations into CLR understandable format. Basically, it provides the Language Interoperability. Language Interoperability means to provide the execution support to other programming languages also in .NET framework.

Language Interoperability can be achieved in two ways :

Managed Code: The MSIL code which is managed by the CLR is known as the Managed Code. For managed code CLR provides three .NET facilities:

* CAS(Code Access Security)
* Exception Handling
* Automatic Memory Management.

Unmanaged Code: Before .NET development the programming language like .COM Components & Win32 API do not generate the MSIL code. So these are not managed by CLR rather managed by Operating System which is called unmanaged code.

COMMON TYPE SYSTEM (CTS):

* Every programming language has its own data type system, so CTS is responsible for the understanding all the data type system of .NET programming languages and converting them into CLR understandable format which will be a common format.

There are 2 Types of CTS that every .NET programming language have :

Value Types: Value Types will directly store the value directly into the memory location. These types work with stack mechanism only. CLR allots memory for these at Compile Time.

Reference Types: Reference Types will contain a memory address of value because the reference types won’t store the variable value directly in memory. These types work with Heap mechanism. CLR allots memory for these at Runtime.

GARBAGE COLLECTOR:

* It is used to provide the Automatic Memory Management feature. Suppose if there is no garbage collector then programmers have to write the memory management codes which will be a kind of overhead on programmers.

JIT(JUST IN TIME COMPILER):

* It is responsible for converting the CIL(Common Intermediate Language ) into machine code or native code using the Common Language Runtime environment.

BENEFITS OF CLR:

* It improves the performance by providing a richly interact between programs at the run time.
* Enhance portability by removing the need of recompiling a program on any operating system that support it.
* Security also increases as it analyze the MSIl instructions whether they are safe or unsafe. Also, the use0 of delegates in place of function pointers enhance the type safety and security.
* Support automatic memory managment with the help of Garbage Collector.
* Provides cross language integration because CTS inside CLR provides a common standard that activate the different languages to extend and share each other’s libraries.
* Provides support to use the components that developed in other .NET programming languages.
* Provide language, platform, and architecture independency.
* It allows the creation of the scalable and multithreaded applications in a easier way as developer has no need to think about the memory management and security isssues.

**Is .NET application platform dependent & Language independent or platform independent?**

The combination of *Operating System Architecture and CPU Architecture* is known as the platform. Platform dependent means the programming language code will run only on particular Operating System. A *.NET application is platform dependent* because of the .NET framework which is only able to run on the Windows-based operating system. The .Net application is platform independent also because of *Mono framework*. Using Mono framework the .Net application can run on any Operating System including windows. Mono framework is a third party software developed by [Novell Company](https://www.microfocus.com/novell/) which is now a part of [Micro Focus Company](https://www.microfocus.com/novell/). It is a paid framework.

**Release History of .NET Framework and its comptability with the different Windows version**

| **.NET VERSION** | **CLR VERSION** | **DEVELOPMENT TOOL** | **WINDOWS SUPPORT** |
| --- | --- | --- | --- |
| **1.0** | 1.0 | Visual Studio .NET | XP SP1 |
| **1.1** | 1.1 | Visual Studio .NET 2003 | XP SP2, SP3 |
| **2.0** | 2.0 | Visual Studio 2005 | N/A |
| **3.0** | 2.0 | Expression Blend | Vista |
| **3.5** | 2.0 | Visual Studio 2008 | 7, 8, 8.1, 10 |
| **4.0** | 4 | Visual Studio 2010 | N/A |
| **4.5** | 4 | Visual Studio 2012 | 8 |
| **4.5.1** | 4 | Visual Studio 2013 | 8.1 |
| **4.5.2** | 4 | N/A | N/A |
| **4.6** | 4 | Visual Studio 2015 | 10 v1507 |
| **4.6.1** | 4 | Visual Studio 2015 Update 1 | 10 v1511 |
| **4.6.2** | 4 | N/A | 10 v1607 |
| **4.7** | 4 | Visual Studio 2017 | 10 v1703 |
| **4.7.1** | 4 | Visual Studio 2017 | 10 v1709 |
| **4.7.2** | 4 | Visual Studio 2017 | 10v 1803 |

**Important Points:**

* Visual Studio is the development tool which is used to design and develop the .NET applications. For using Visual Studio, the user has to first install the .NET framework on the system.
* In the older version of Windows OS like XP SP1, SP2 or SP3, .NET framework was integrated with the installation media.
* Windows 8, 8.1 or 10 do not provide a pre-installed version 3.5 or later of .NETFramework. Therefore, a version higher than 3.5 must be installed either from a Windows installation media or from the Internet on demand. Windows update will give recommendations to install the .NET framework.