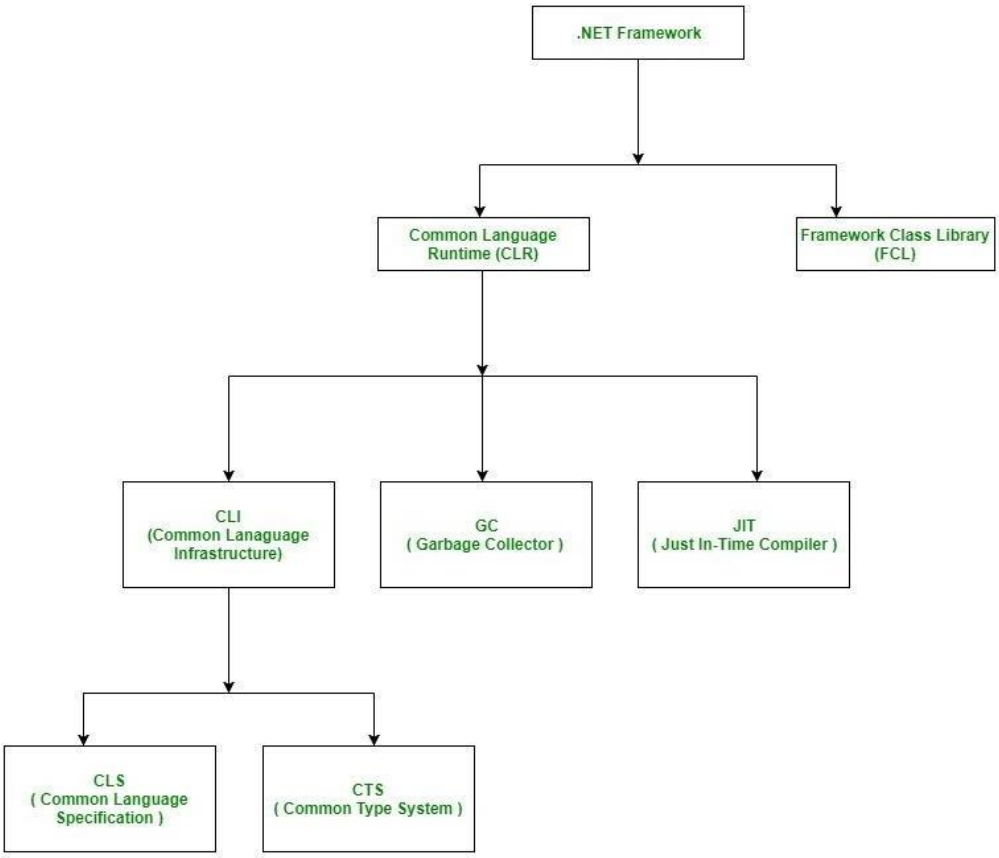
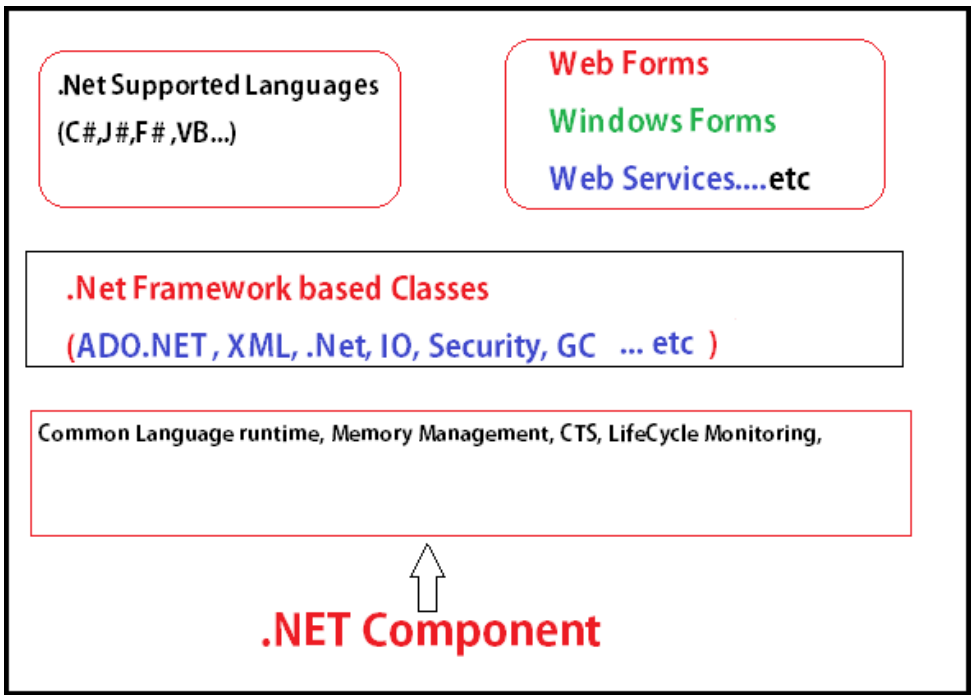
# .net-framework-components:

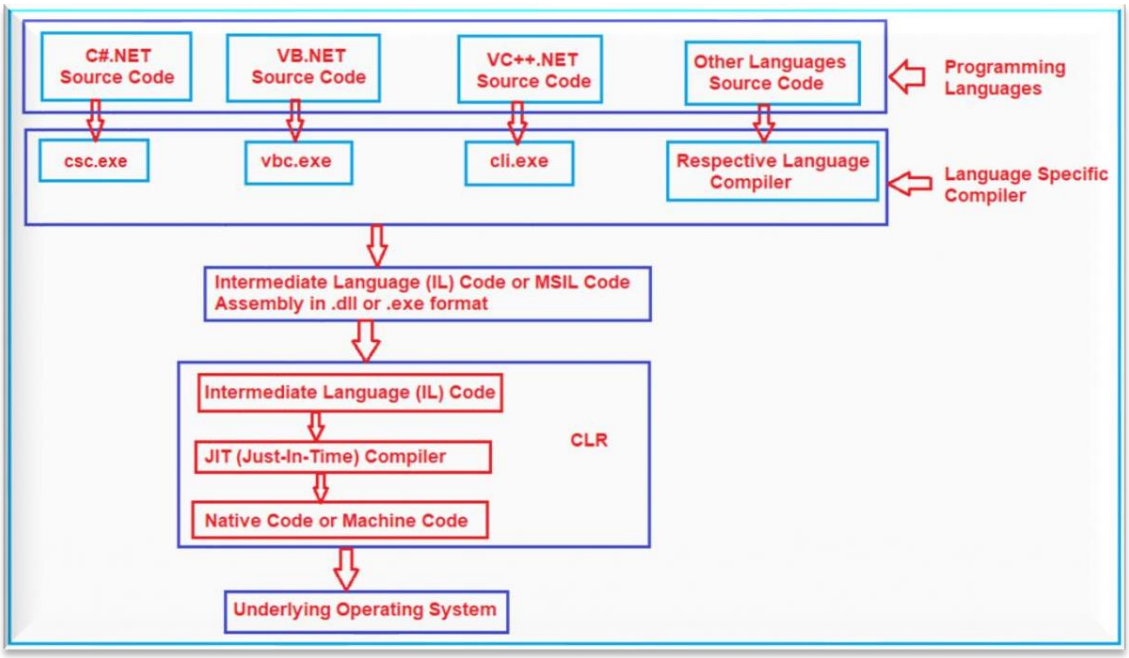



".Net is a S/W framework that includes everything that require for developing application like web, desktop, mobile & IoT. Dot Net framework comes with a single class library so weather you write you code in VB, C# or J# is doesn't matter."

Feature of .Net:

- ✓ Platform Neutral or platform independent
- ✓ It is a layer between OS and Programming Language
- ✓ It supports many programming language. Such as C#,J#,F# and VB
- ✓ .Net provides a common set of class Libraries which can be accessed from any .net based programming language


.Net Supported Languages
(C#,J#,F#,VB...)
Web Forms
Windows Forms
Web Services....etc
.Net Framework based Classes
(ADO.NET, XML, .Net, IO, Security, GC ... etc )
Common Language runtime, Memory Management, CTS, LifeCycle Monitoring,
.NET Component


C#.NET Source Code
VB.NET Source Code
VC++.NET Source Code
Other Languages Source Code
Programming Languages
csc.exe
vbc.exe
cli.exe
Respective Language Compiler
Language Specific Compiler
Intermediate Language (IL) Code or MSIL Code
Assembly in .dll or .exe format
Intermediate Language (IL) Code
JIT (Just-In-Time) Compiler
Native Code or Machine Code
CLR
Underlying Operating System

# What is CLR

***Common Language Runtime***. CLR working is similar to JVM in Java. It is runtime environment that converts MSIL code into Machine code or Binary code or Low Level Language which easily understandable and executable by the CPU or machine.

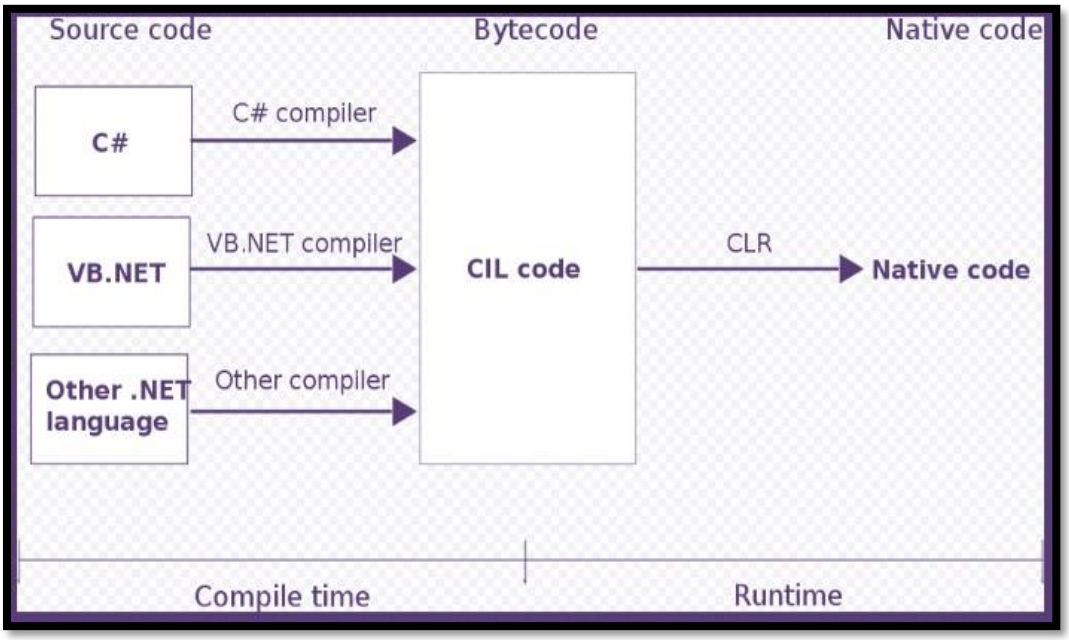
## CLR provide number of services that includes:

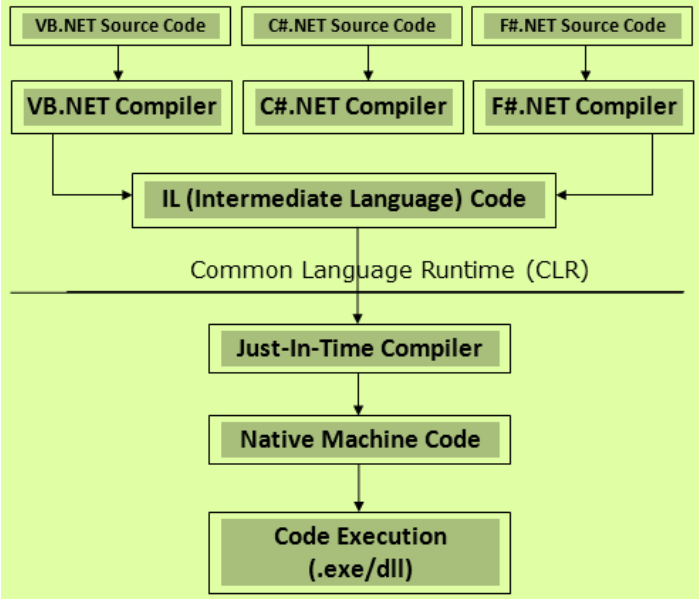
- ✓ Loading & execution of .net code
- ✓ Memory isolation & Management for application
- ✓ Verification of type safety
- ✓ Compilation of intermediate Language (MSIL code) into native executable code.
- ✓ Automatic Garbage collection
- ✓ Security
- ✓ Interoperability
- ✓ Managing execution and occurrence of error and
- ✓ Provides supports for debugging

In .NET, the code is compiled twice.

1. In 1st compilation source code (High-Level Code) is compiled by the respective language compiler and the language compiler generates intermediate code which is also known as MSIL (Microsoft Intermediate Language) or IL (Intermediate language code) Or Managed code.
2. In the 2nd compilation, MSIL is converted into Native Code (Machine code) using CLR.

Always 1st compilation is slow and 2nd compilation is first.






## .NET CLR Functions

Following are the functions of the CLR.

- ✓ It converts the program into native code.
- ✓ Handles Exceptions
- ✓ Provides type-safety
- ✓ Memory management
- ✓ Provides security
- ✓ Improved performance
- ✓ Language independent
- ✓ Platform independent
- ✓ Garbage collection
- ✓ Provides language features such as inheritance, interfaces, and overloading for object-oriented programming.

The CLR takes the IL (Intermediate Language) code and gives it to something called JIT (Just-in-Time) Compiler. The JIT compiler reads each and every line of the IL code and converts it to machine-specific instructions (i.e. into binary format) which can be executed by the underlying Operating System.

### What is MSIL:

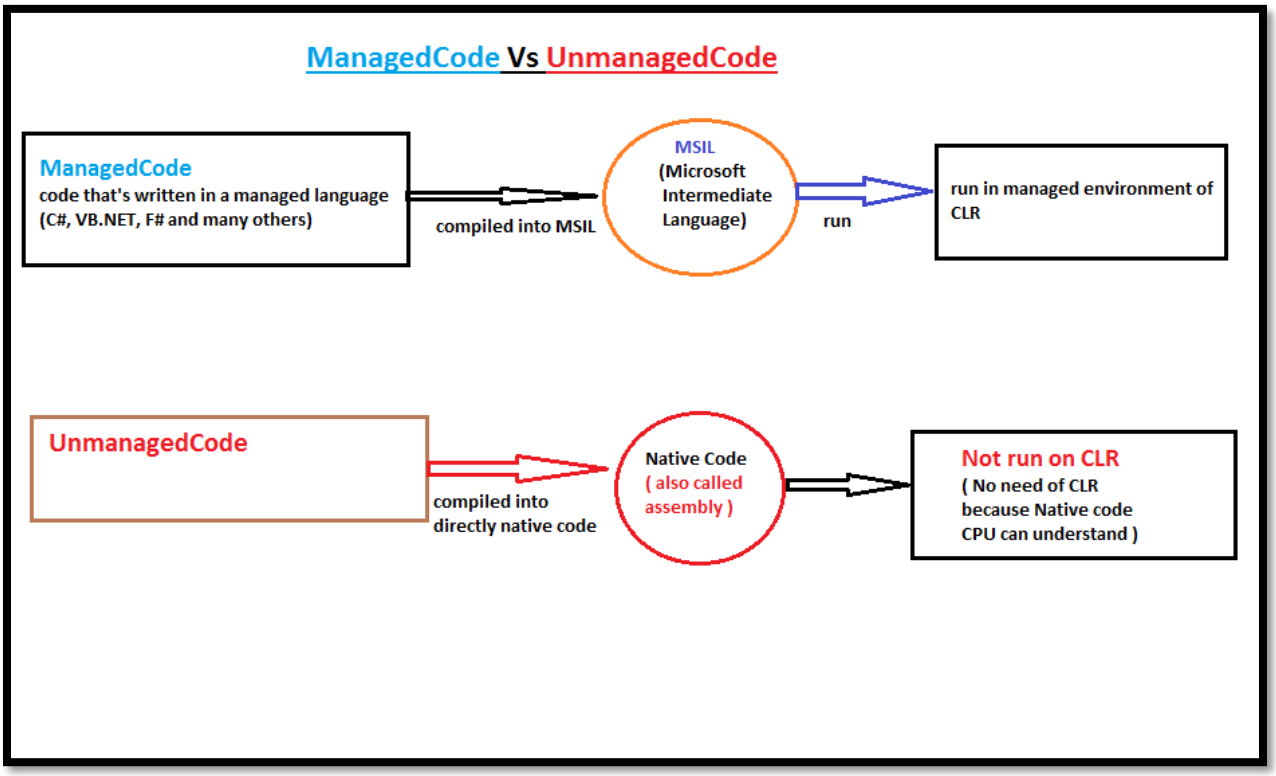
MSIL stands for Microsoft Intermediate Language. A .net programming language does not compile its source code directly into the executable machine code instead of compile it into an intermediate code called MSIL.

A .net Language source code is first compile to MSIL and Send it to CLR by which it is converted into the machine code. MSIL is similar to Javabyte code.

# ManagedCode:

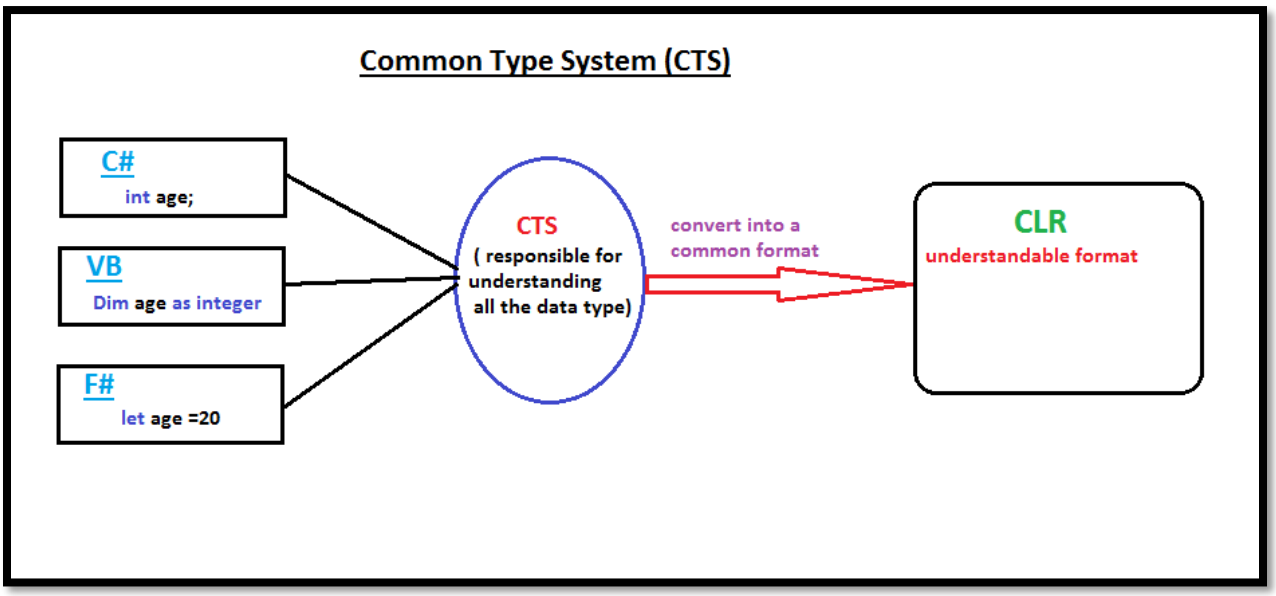
The codes which run under the complete control of CLR are called Managed Code in .NET. These kinds of code are run by dot net runtime environment. If the dot net framework is not installed or if dot net runtime is not available, then these kinds of codes are not going to be executed. CLR will provide all the facilities and features of .NET to the managed code execution like Language Interoperability, Automatic memory management, Exception handling mechanism, code access security, etc. Code written using the .net framework is managed when it is executed.

This stage is usually referred as runtime. This means that the CLR loads after or application by managing memory handling security allowing cross language debugging and so:



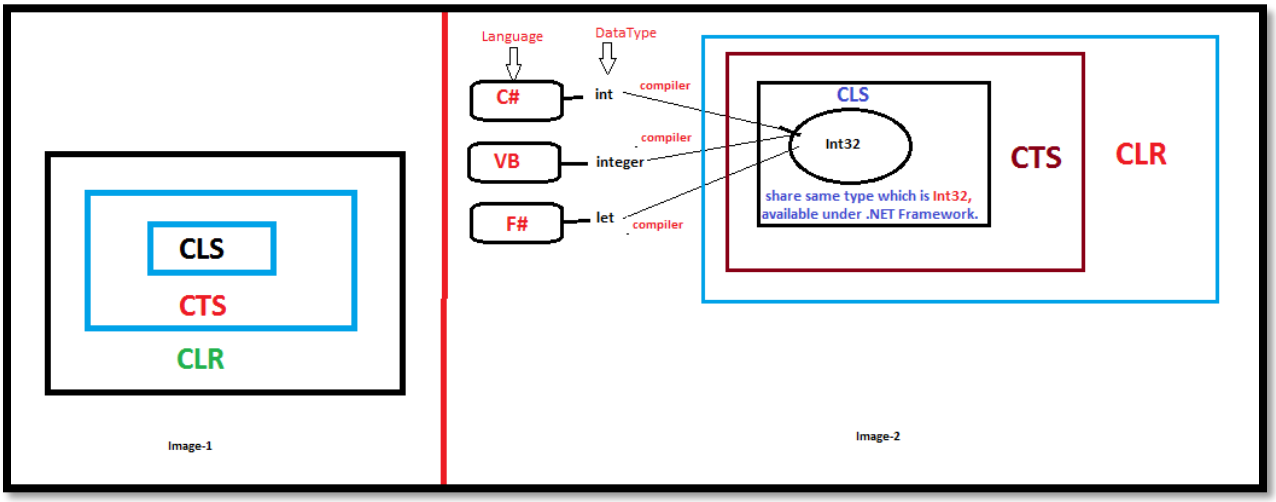
## What is the Common Type System in .NET Framework?

The .NET Framework supports many programming languages such as C#, VB.NET, J#, etc. Every programming language has its own data type. One programming language data type cannot be understood by other programming languages. But, there can be situations where we need to communicate between two different programming languages. For example, we need to write code in the VB.NET language and that code may be called from C# language. In order to ensure smooth communication between these languages, the most important thing is that they should have a Common Type System (CTS) which ensures that data types defined in two different languages get compiled to a common data type.



## What is Common Language Specification (CLS) in .NET Framework?

CLS (Common Language Specification) is a part of CLR in the .NET Framework. The .NET Framework supports many programming languages such as C#, VB.NET, J#, etc. Every programming language has its own syntactical rules for writing the code which is known as a language specification. One programming language syntactical rules (language specification) cannot be understood by other programming languages. But, there can be situations where we need to communicate between two different programming languages. In order to ensure smooth communication between different .NET Supported Programming Languages, the most important thing is that they should have Common Language Specifications which ensures that language specifications defined in two different languages get compiled to a Common Language Specification.




it is a subset of CTS. It defines a set of rules and restrictions that every language must follow which runs under the .NET framework.

| |
|---|
| Source code------ → User |
| MSIL ---------- → Language compiler |
| Native code ------- → JIT |
| Managed code---- → CLR |