**What is Microsoft .Net Framework?**

The .Net framework is a software development platform developed by Microsoft. The framework was meant to create applications, which would run on the Windows Platform. The first version of the .Net framework was released in the year 2002.

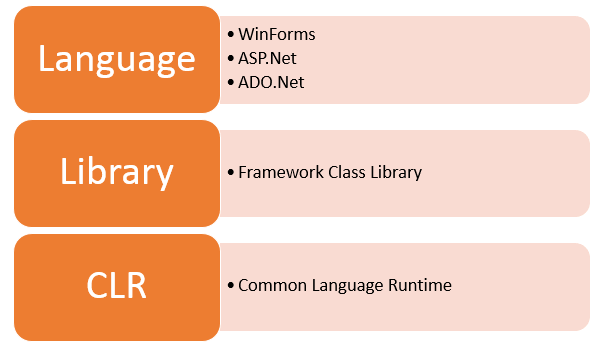
The version was called .Net framework 1.0. The .Net framework has come a long way since then, and the current version is 4.7.1.

The .Net framework can be used to create both - **Form-based** and **Web-based** applications. [Web services](https://www.guru99.com/web-services-tutorial.html) can also be developed using the .Net framework.

The framework also supports various programming languages such as Visual Basic and C#. So developers can choose and select the language to develop the required application. In this chapter, you will learn some basics of the .Net framework.

**.Net Framework Architecture**

The basic architecture of the .Net framework is as shown below.

[](https://cdn.guru99.com/images/c-sharp-net/052416_1343_WhatisNETFr1.png)

.net framework architecture diagram

**.NET Components**

The architecture of the .Net framework is based on the following key components;

**1. Common Language Runtime**

The "Common Language Infrastructure" or CLI is a platform on which the .Net programs are executed.

The CLI has the following key features:

* Exception Handling - Exceptions are errors which occur when the application is executed.

Examples of exceptions are:

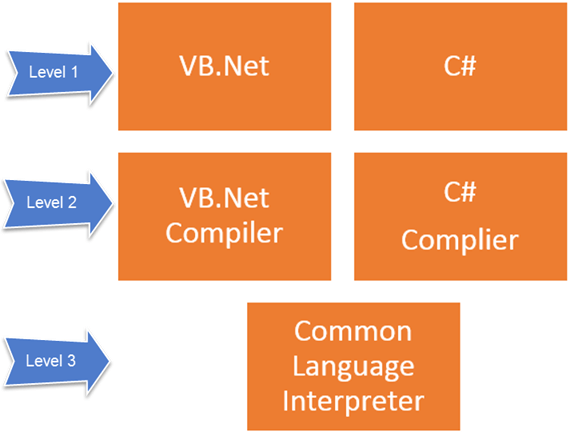
* + If an application tries to open a file on the local machine, but the file is not present.
  + If the application tries to fetch some records from a database, but the connection to the database is not valid.
* Garbage Collection - Garbage collection is the process of removing unwanted resources when they are no longer required.

Examples of garbage collection are

* + A File handle which is no longer required. If the application has finished all operations on a file, then the file handle may no longer be required.
  + The database connection is no longer required. If the application has finished all operations on a database, then the database connection may no longer be required.
* Working with Various programming languages –

As noted in an earlier section, a developer can develop an application in a variety of .Net programming languages.

1. Language - The first level is the programming language itself, the most common ones are VB.Net and C#.
2. Compiler – There is a compiler which will be separate for each programming language. So underlying the VB.Net language, there will be a separate VB.Net compiler. Similarly, for C#, you will have another compiler.
3. Common Language Interpreter – This is the final layer in .Net which would be used to run a .net program developed in any programming language. So the subsequent compiler will send the program to the CLI layer to run the .Net application.

[](https://cdn.guru99.com/images/c-sharp-net/052416_1343_WhatisNETFr2.png)

**2. Class Library**

The .NET Framework includes a set of standard class libraries. A class library is a collection of methods and functions that can be used for the core purpose.

For example, there is a class library with methods to handle all file-level operations. So there is a method which can be used to read the text from a file. Similarly, there is a method to write text to a file.

Most of the methods are split into either the System.\* or Microsoft.\* namespaces. (The asterisk \* just means a reference to all of the methods that fall under the System or Microsoft namespace)

A namespace is a logical separation of methods. We will learn these namespaces more in detail in the subsequent chapters.

**3. Languages**

The types of applications that can be built in the .Net framework is classified broadly into the following categories.

* WinForms – This is used for developing Forms-based applications, which would run on an end user machine. Notepad is an example of a client-based application.
* ASP.Net – This is used for developing web-based applications, which are made to run on any browser such as Internet Explorer, Chrome or Firefox.
  + The Web application would be processed on a server, which would have Internet Information Services Installed.
  + Internet Information Services or IIS is a Microsoft component which is used to execute an [Asp.Net](https://www.guru99.com/asp-net-tutorial.html) application.
  + The result of the execution is then sent to the client machines, and the output is shown in the browser.
* ADO.Net – This technology is used to develop applications to interact with Databases such as Oracle or Microsoft [SQL](https://www.guru99.com/sql.html) Server.

Microsoft always ensures that .Net frameworks are in compliance with all the supported Windows operating systems.

**.Net Framework Design Principle**

The following design principles of the .Net framework is what makes it very relevant to create .Net based applications.

1. Interoperability - The .Net framework provides a lot of backward support. Suppose if you had an application built on an older version of the .Net framework, say 2.0. And if you tried to run the same application on a machine which had the higher version of the .Net framework, say 3.5. The application would still work. This is because with every release, Microsoft ensures that older framework versions gel well with the latest version.
2. Portability- Applications built on the .Net framework can be made to work on any Windows platform. And now in recent times, Microsoft is also envisioning to make Microsoft products work on other platforms, such as iOS and Linux.
3. Security - The .NET Framework has a good security mechanism. The inbuilt security mechanism helps in both validation and verification of applications. Every application can explicitly define their security mechanism. Each security mechanism is used to grant the user access to the code or to the running program.
4. Memory management - The Common Language runtime does all the work or memory management. The .Net framework has all the capability to see those resources, which are not used by a running program. It would then release those resources accordingly. This is done via a program called the "Garbage Collector" which runs as part of the .Net framework.

The garbage collector runs at regular intervals and keeps on checking which system resources are not utilized, and frees them accordingly.

1. Simplified deployment - The .Net framework also have tools, which can be used to package applications built on the .Net framework. These packages can then be distributed to client machines. The packages would then automatically install the application.

**Summary**

* .Net is a programming language developed by Microsoft. It was designed to build applications which could run on the Windows platform.
* The .Net programming language can be used to develop Forms based applications, Web based applications, and Web services.
* Developers can choose from a variety of programming languages available on the .Net platform. The most common ones are VB.Net and C#.

# Code-Behind Vs Inline Code in Asp.Net

## What is Code Behind?

Code Behind refers to code for **ASP.NET** page which is contained within a separate class file. It is composed in a different class record that can have the extension of **.aspx.cs** or **.aspx.vb** relying upon the language used. It allows a clean separation of HTML from the presentation logic. In the **code-behind** file, you create a class (which can be any class derived from the Page class) that serves as the base class for the web page you create in the .aspx file. This relationship between your class and the web page is established by a **Page directive** at the top of the .aspx file:



The inherits attribute identifies the class created in the **code-behind** file from which this .aspx file will derive. One major point of Code Behind is that the code for all the Web pages is compiled into a **DLL file** that allows the web pages to be hosted free from any Inline Server Code.

## What is Inline Code?

Inline Code is embedded directly within the **ASP.NET** page that has an extension of **.aspx** . It permits the code to be composed along with the HTML source code using a < Script > tag. When the page is deployed, the source code is deployed along with the Web **Forms page** , because it is physically in the .aspx file. However, you do not see the code, only the results are rendered when the page runs.

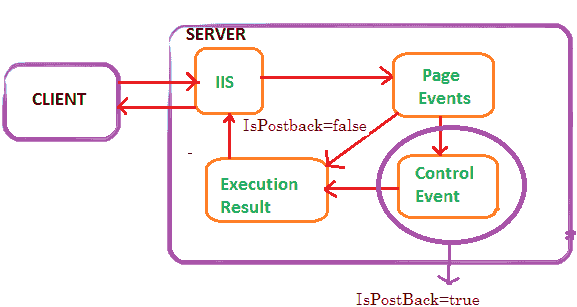
# What is PostBack

Postback is actually sending all the information from client to web server, then web server process all those contents and returns back to the client.

**IsPostBack**

The IsPostBack property can be used to determine if the page is submitted to itself. When a form is submitted back to the **same** page that contains it, it's called a post back. ASP.NET provides a property called IsPostBack that is TRUE when the page is being loaded as a result of a post back, and is **FALSE** otherwise

Most of the time ASP control will cause a post back (e. g. buttonclick) but some don't unless you tell them to do In certain events ( Listbox Index Changed,RadioButton Checked etc..) in an ASP.NET page upon which a PostBack might be needed.



IsPostBack is a property of the Asp.Net page that tells whether or not the page is on its initial load or if a user has perform a button on your web page that has caused the page to post back to itself. The value of the Page.IsPostBack property will be set to true when the page is executing after a postback, and false otherwise. We can check the value of this property based on the value and we can populate the controls on the page.

IsPostback is normally used on page \_load event to detect if the web page is getting generated due to postback requested by a control on the page or if the page is getting loaded for the first time.

Protected Sub Page\_Load(sender As Object, e As EventArgs)

If Not IsPostBack Then

'generate form

Else 'process submitted data

End If

End Sub

**What is the use of autopostback in asp net?**

**Autopostback** is the mechanism, by which the page will be posted back to the server automatically based on some events in the web controls. In some of the web controls, the property called **autopostback**, which if set to true, will send the request to the server when an event happens in the control.