

[www.jump2learn.com](http://www.jump2learn.com)

Jump2Learn  
PUBLICATION

# .NET PROGRAMMING with GUI



Jump2Learn - The Online Learning Place

Dr. RonakKumar K. Panchal | Mrs. Dipjaya Patel | Dr. Rinkuben N. Patel

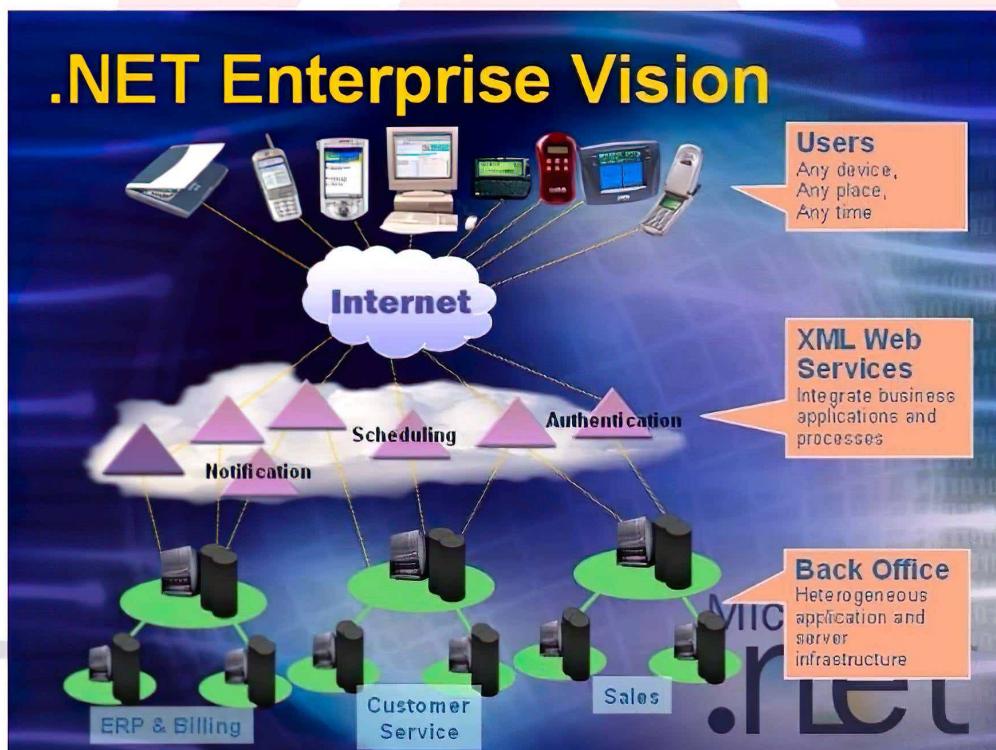
## 1. Overview of Microsoft .NET Programming

### 1.1 The .NET Framework

#### What is .Net Platform?

Microsoft .NET is a software development platform based on virtual machine architecture. Dot Net Platform is:

- **Language Independent** – dot net application can be developed different languages (such as C#, VB, C++, etc.)
- **Platform Independent** – dot net application can be run on any operating system which has .net framework installed.
- **Hardware Independent** – dot net application can run on any hardware configuration It allows us to build windows based application, web based application, web service, mobile application, etc.



### How .Net Address Today's Challenges

Clearly, business users today are faced with a lot of technology, but a limited ability to get at their to get at their data in meaningful, productive way.

### What are the benefits of .Net?

1. Simplify Application Development
2. Simplify Application Development
3. XML everywhere
4. Universal Data Access
5. Web Service: Collaboration over the internet

### The Building Blocks of .Net are:

- The .Net Framework
- .Net Enterprise Servers
- .Net Building Block Services
- Visual Studio .Net

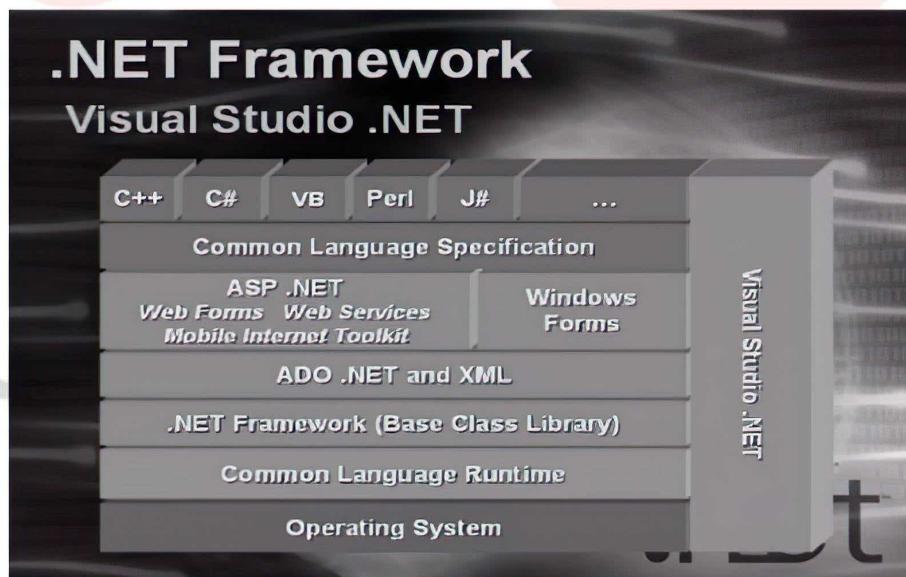
### What is .Net Framework?

.Net framework provided rich set of functionality classes, Assemblies, Data types and services and also simplified application development & deployment.

OR

.Net Framework provides a foundation upon which .net application and xml web services are built and executed.

For developers, Microsoft provides the new .Net Framework, which is a set of system secure class and data type that enhance developer productivity and give easier access to the deep set up of functionality provided by windows. The .Net Framework is shown in the figure bellows:



The .Net Framework is a layered system of classes and service that starts with the operating system service, and moves up through a set of system class (the Base Class Library) and abstracted classes(For example ASP.NET).

- **Common Language Runtime:** A rich runtime environment that handles important runtime tasks for the developer, including memory management and garbage collection. Built around the common Type System and defines a common type system for all language.
- **Base Class Library:** A rich set of functional base class that may be inherited and extended by other classes in Framework. For example, System.Object provides base object functionality that all classes in the Framework inherit. System.IO provides serialization in end from different Input/Output devices, such as files and stream.
- **Extended Class Libraries:** class libraries that are focused on one aspect of development. These classes are extended from the Base Class Library, and are design to make it easier and faster to developed a specific type of application. For example, ASP.NET includes classes that are focused on developing Web Services. Other example, include ADO.NET (for data access), XML.NET in braces to parse and manipulate DOCs) and Windows Forms (the successor to VB forms).
- **Common Language Specification:** define requirements for .Net Languages, by specifying a set of rules that .Net compliant Languages must follow. One of these rules in that the language must ad here to common type system.
- **Multiple Programming Language:** VB.NET, C#.NET and C++.NET are just some of the many languages that are available for coding in .Net. The .Net Framework provides one platform and unified programming model for several languages. Java is conspicuously absent from the .Net family of language, probably due to the licensing dispute between Sun Microsystems and Microsoft.
- **Visual Studio .Net:** an integrated development environment for coding with the .Net Framework. The diagram shows VS.NET spanning the entire .Net Framework because it provides tools that access each part of the Framework.
- **Windows and COM+ Service:** There are technically not part of the .NET Framework, but they are a requirement for today's .NET Framework SDK.

To summarize, the important concept behind the .NET Framework are:

- Built on a common set of Framework classes
- Provides a Common Type System, that is the cornerstone of unified programming model for all .NET compliant languages
- Provides a Common Language Runtime that provides runtime service for components and applications
- Provides extended class libraries for ASP.NET, ADO.NET, XML.NET and Windows Forms
- Visual Studio.NET is an integrated development environment for the NET Framework

### The .NET Enterprise Servers:

Microsoft is orienting all of their recent and upcoming technology around .NET. To this end they have identified a suite of products called **.NET Enterprise Servers**, which are server-based application that web enable enterprise systems. These include applications that you may already be using but did not realize were part of .NET initiative. Examples of .NET Enterprise Servers include:

- Windows 2000 Advance Server
- Application Center 2000/2008
- SQL Server 2000/2005/2008
- Exchange Server 2000
- Host Integration Server 2000
- Internet Security and Acceleration Server 2000/2008
- Commerce Server 2000
- BizTalk Server

### .NET Building Block Services

The .NET building block services will include:

- Authentication
- Notification and Messaging
- Directory and Search
- Calendar
- XML Store

### Visual Studio .NET

Visual Studio .Net is newest version of Microsoft's development toolkit for creating .NET solutions. It is designed to promote **Rapid Application Development (RAD)**. The .NET Framework SDK actually provides everything that you need. However, you will miss out on many of the benefits that Visual Studio.NET provides: an integrated development environment and tight integration with the .NET Framework.

### The key features of Visual Studio.NET are:

- Full integration with the .NET Framework
- Integrated development environment
- Mixed Language development including cross language debugging
- RAD features for application development
- Visual Designers for XML, HTML and Data
- Expanded debugging across projects, including stored procedures

## Overview of .NET Applications

There are several types of applications that you can build with .NET:

- **Windows Forms Applications:** Windows Form Applications are the newest generation of the traditional windows-based applications that provide a form-based user interface and n-Tier, partitioned architecture. Windows Forms are objects that are derived from the .NET Framework.

**Windows Forms provide the following useful feature:**

- **A new Forms Architecture:** an object oriented set of classes including the base Forms class
- **The Control object Model:** a set of Windows Controls for the user interface
- **A new Event Model:** A set of events based on delegate which are similar to callbacks
- **Windows Forms Controls:** Windows Form Controls are the successors to ActiveX controls. They are reusable components that provide a user interface and responsive to user events
- **Windows Service Application:** Windows Service applications were formerly known as NT services. They are executables that run in independent windows sessions with no user interaction. Microsoft developers will be most familiar with the following services:
  - Distributed Transaction Coordinator
  - IIS Admin Service
  - Simple Mail Transport Protocol (SMTP)
  - Task Scheduler
  - Windows Installer
  - World Wide Web publishing Service

- **ASP.NET Web Applications:** ASP.NET is the next generation platform for developing web application.

ASP.NET provides the following two programming models:

- **Web Forms:** these are analogous to Windows Forms, and even provide Web Controls that can be dropped on to the form to provide a user interface, and to automate common functions the functionality of client – side script.
- **Web Services:** these are remote application components that receive and respond to requests using open standard protocols, RPC calls over HTTP using XML (combined into SOAP envelopes).
- **Web Services:** ASP.NET and the .NET Framework together provide classes and services for building web services components.
- **Web Simple Description Language (WSDL)** allows outside consumers to gather the information they need to communicate with your web service.
- **Simple Object Access Protocol (SOAP):** The SOAP specification defines how to send XML over HTTP. Requests and responses to and from Web Services are formatted and passed via SOAP envelopes.

## 1.2 The Common Language Runtime (CLR):

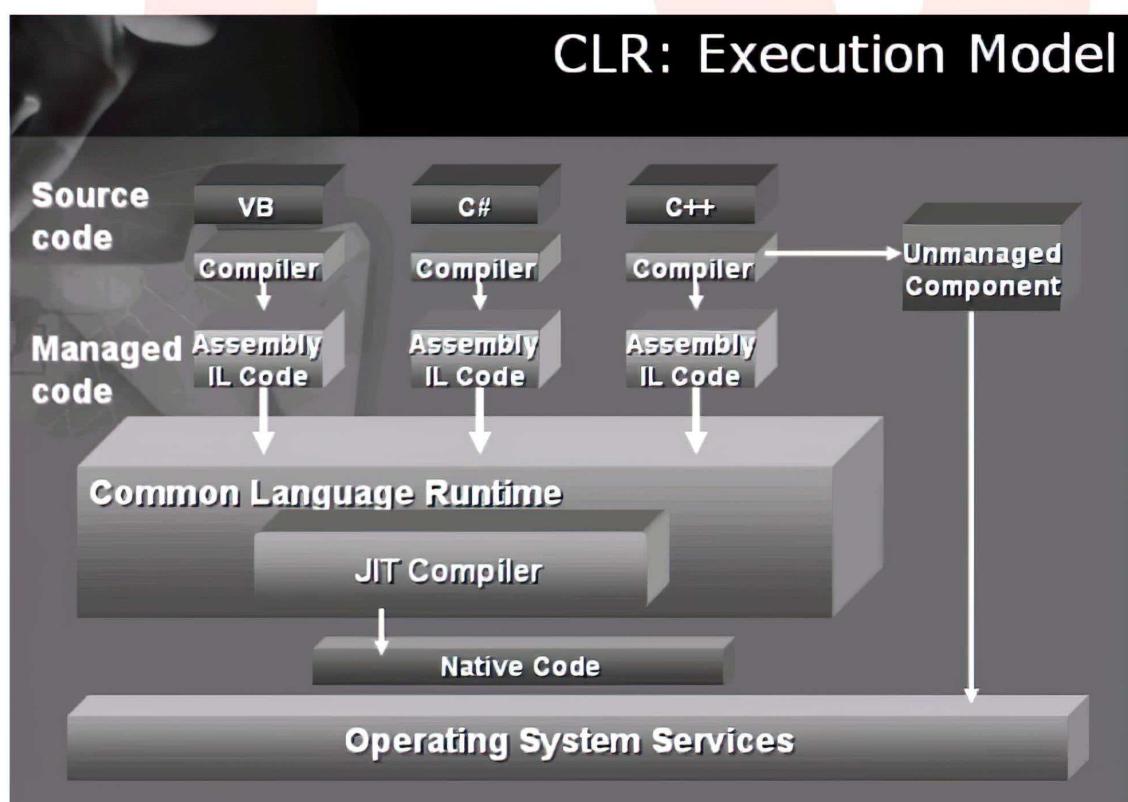
### Definition:

Common Language Runtime (CLR) is execution engine of .Net Framework based application. Code that runs under the control of the CLR is called **managed code**, because the CLR defines the rules that code's development language must conform to specifically, the CLR works with two other Framework services to define the rules for .NET languages. These are:

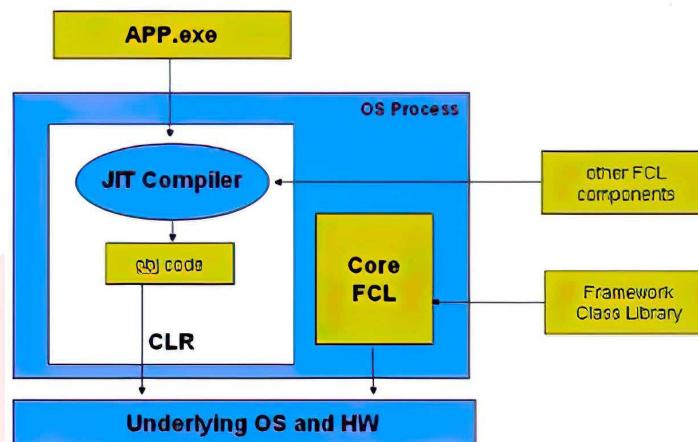
- **Common Type System (CTS)**: defines standard reference and value types that are supported in the .NET Framework
- **Common Language Specification (CLS)**: defines rules that a development must comply with in order to be managed by the .NET Framework

### The Specific benefits of the .NET Framework are:

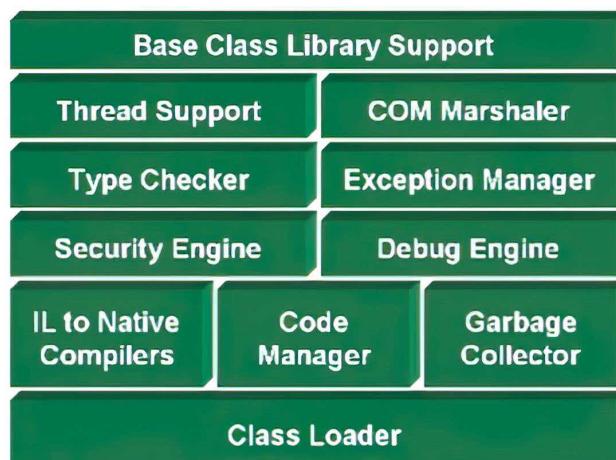
- DLL “Heaven” not “DLL Hell”
- Component integration replaces interface
- Simplified deployment
- Improved resource management
- Multiple language integration
- Unified, extensible Class Library
- Structured exception handling



The Common Language Runtime will manage code for any language that conforms to the CLS and the CTS. .NET code is compiled in a two-step process whereby the code is first converted into a language neutral generalized instruction set called Microsoft Intermediate Language (IL). Next, the Just-In-Time executed by the CLR. .NET code is compiled into assemblies, which are similar to dynamic link libraries (DLLs) except that they hold Meta data and self-describing.



### Common Language Runtime



The CLR provides a number of runtime support service using **Virtual Execution System (VES)**. The VES is responsible for implementing and enforcing the Common Type System. The execution engine uses Meta data information to understand the structure of the components.

The specific components of the VES are:

- Class Loader (Load managed Code)
- Microsoft Intermediate Language (MSIL)
- MSIL-to-Native code conversion
- Verification of Type safety, according to CTS
- Stack Walker
- Memory Management and Garbage Collection
- Profiling and Debugging
- Co-Instance Execution
- Unmanaged Code

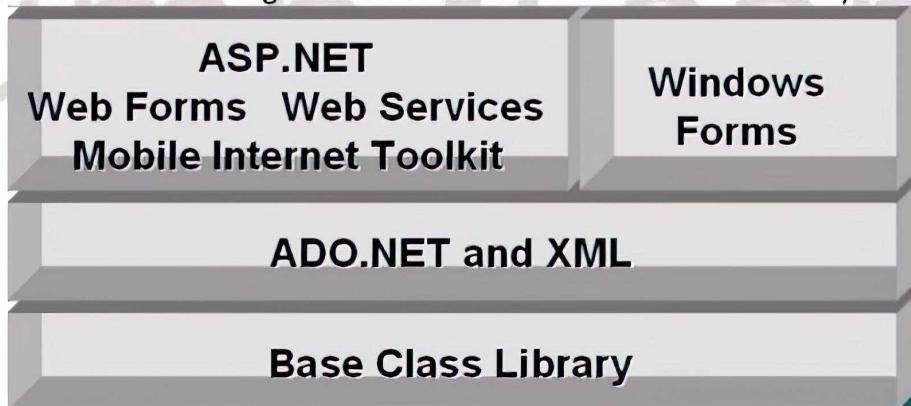
### 1.3 The .NET Framework Class Library:

The .Net Class Framework provides developers with object-oriented, extensible classes, interfaces and types for accessing system functionality. The Class Framework is organized into hierarchical libraries of classes that may be used consistently across any .Net compliant language.

The .Net Class Framework overcomes these limitations in the following ways:

- **Namespaces:** Classes, Interfaces and Types are organized into hierarchical structures called namespaces which group related classes and keep groups of classes distinct.
- **Unified Programming Framework:** .Net provides a Common Type System that standardizes data types across the Framework, Which puts all languages on an equal footing in terms of what data types they can communicate with. There are some differences between languages but in general they are all able to access the same classes.
- **Object-Oriented:** The Class Framework provides extensible classes that may be manipulated using standard object oriented operations including inheritance, method overriding and polymorphism.

The figure below shows the organization of the .Net Framework Class Library at high level:



The **System** namespace is the root namespace for all other namespaces in the .NET Framework.

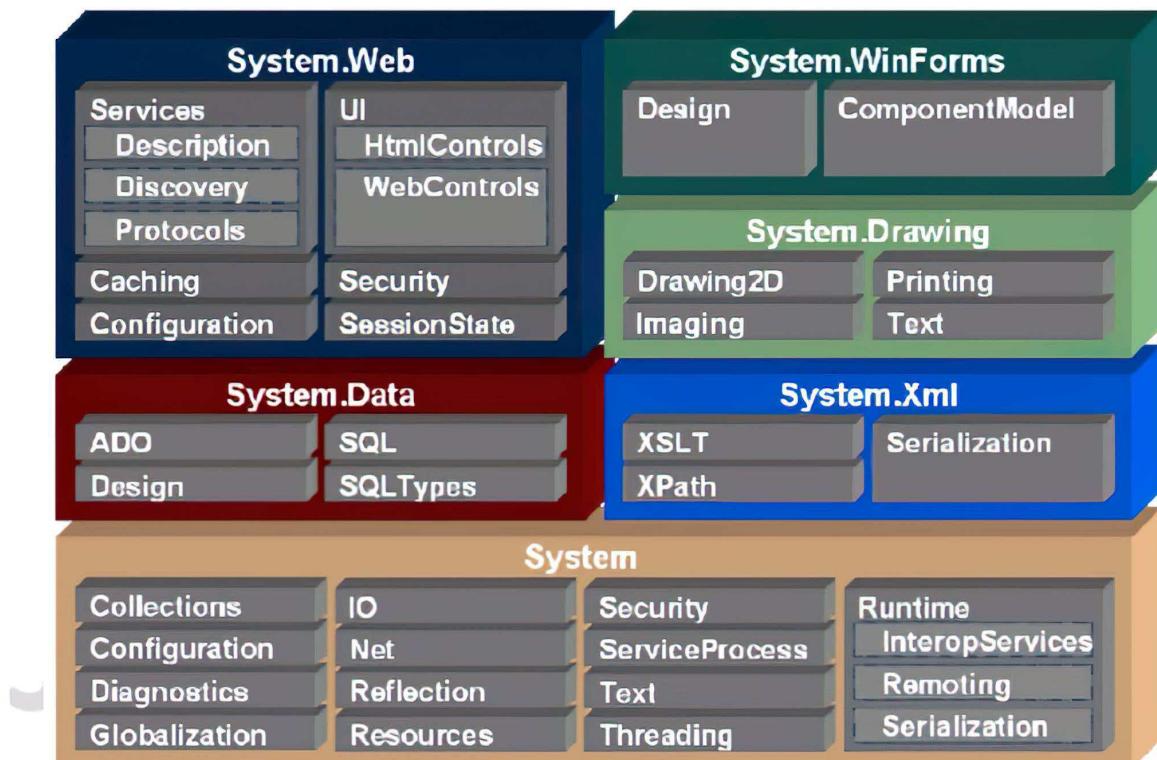
The important classes in the System namespace are:

- **System.Object**: All classes in the .NET Framework inherit this class. This class ensures that every Framework class implements a basic standard interface.
- **System.Exception**: Contains classes that standardize Exception handling

The features in the figure above translate to namespaces in the class library:

- **ADO.NET** provides the System.Data namespace for data access classes
- **ASP.NET** provides the System.Web namespace for ASP.NET code, including control and classes that support Web Services
- **XML.NET** provides the System.Xml namespace for XML classes. It contains other namespaces such as XPath, XSLT and Serialization namespaces
- **Windows Forms** provides the System.Windows.Forms namespace for classes that support windows forms control and functionality

## Some .NET Base Class Libraries



The .NET Class Framework provides object-oriented access to a broad range of functionality, including direct system functionality.