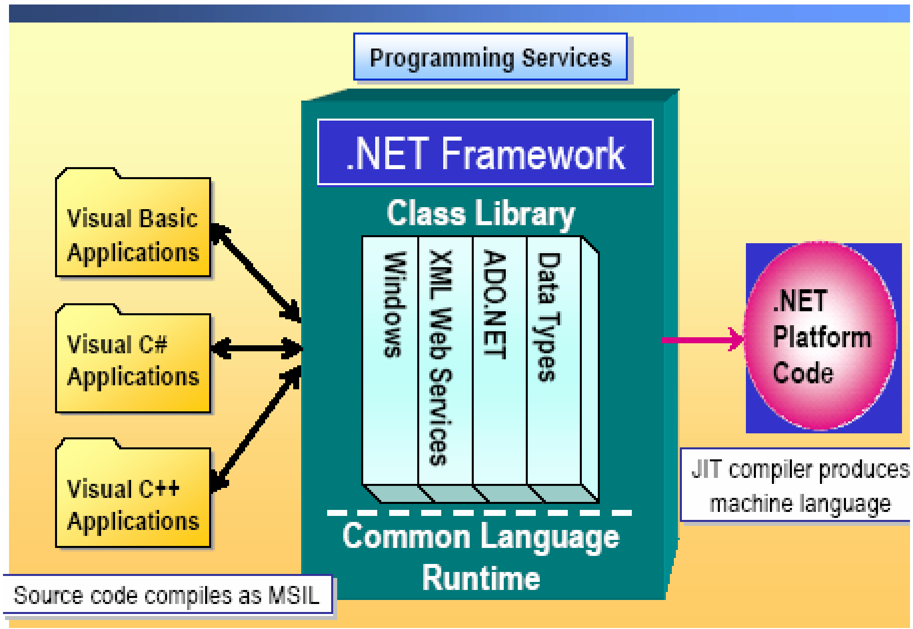
Visual Studio .NET is an integrated development environment that helps you to quickly design, develop, debug, and deploy .NET Framework solutions. You can access a common set of tools, designers, and editors from any Visual Studio .NET programming language. You can create Windows Forms and Web Forms applications that integrate data and business logic.

The .NET Framework



The .NET Framework is a set of programming services designed to simplify application development in the highly distributed environment of the Internet. The .NET Framework has two main components: the common language runtime and the class library.

Common Language Runtime

The *common language runtime* is the foundation of the .NET Framework. In the .NET environment, programmers develop applications in the .NET compatible language of their choice, the code is compiled into MSIL, and the runtime manages and runs the compiled code. CLR is necessary for all .NET languages to run.

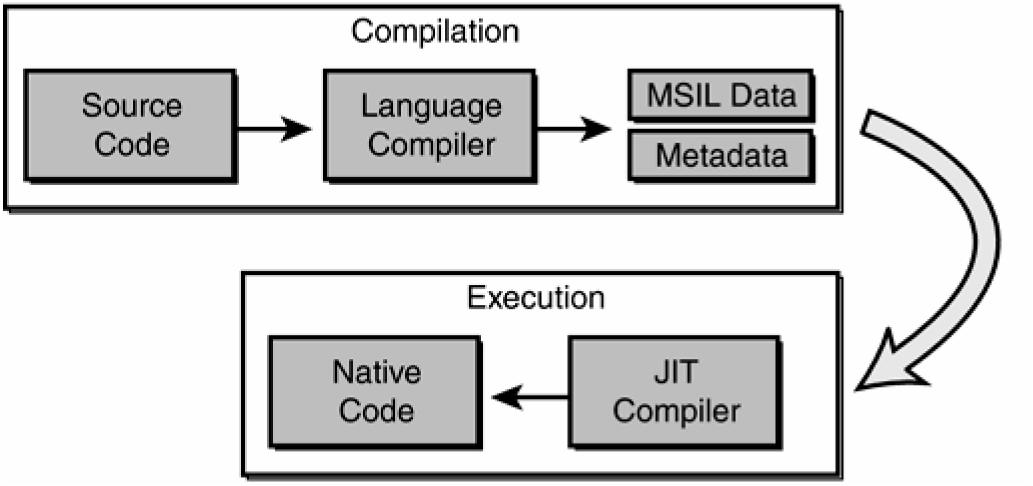
Note: **The .NET compatible programming languages** like VB.NET, Visual C#., Visual C++, Jscript, and visual J#

Class library

The class library is a comprehensive object-oriented collection of reusable classes that are organized by their functionality into hierarchical namespaces. You can use the class library to develop applications ranging from traditional client applications to applications based on the latest innovations provided by Microsoft ASP.NET and XML Web services.

When you compile the application, it is translated into the run time's common language, Microsoft intermediate language (MSIL). After the application is compiled, the runtime manages the execution of the application.

Figure 1.5. The JIT process and verification.



The runtime includes a feature called just-in-time (JIT) compilation that translates the MSIL code into the machine language (Native Code) of the system on which the application will run. When a client device on the .NET platform launches the Visual Basic .NET application, it starts running in the machine language of the client system and can fully integrate and interact with other .NET-based applications and services regardless of the language in which it was developed.

Class

Class is a blueprint or template for an object. In the next lectures you'll study more about the Class and object. Till now you have to know that:
A *class* is a named programming entity that comprises a common set of methods, properties, and attributes. For example, **Form** is one of the classes in the **System.Windows.Forms** namespace that is used to create Windows Forms.

Namespace

A *namespace* identifies a collection of related classes and/or other namespaces in the .NET Framework. Examples of namespaces include the following:

- **System**
- **System.Windows.Forms**

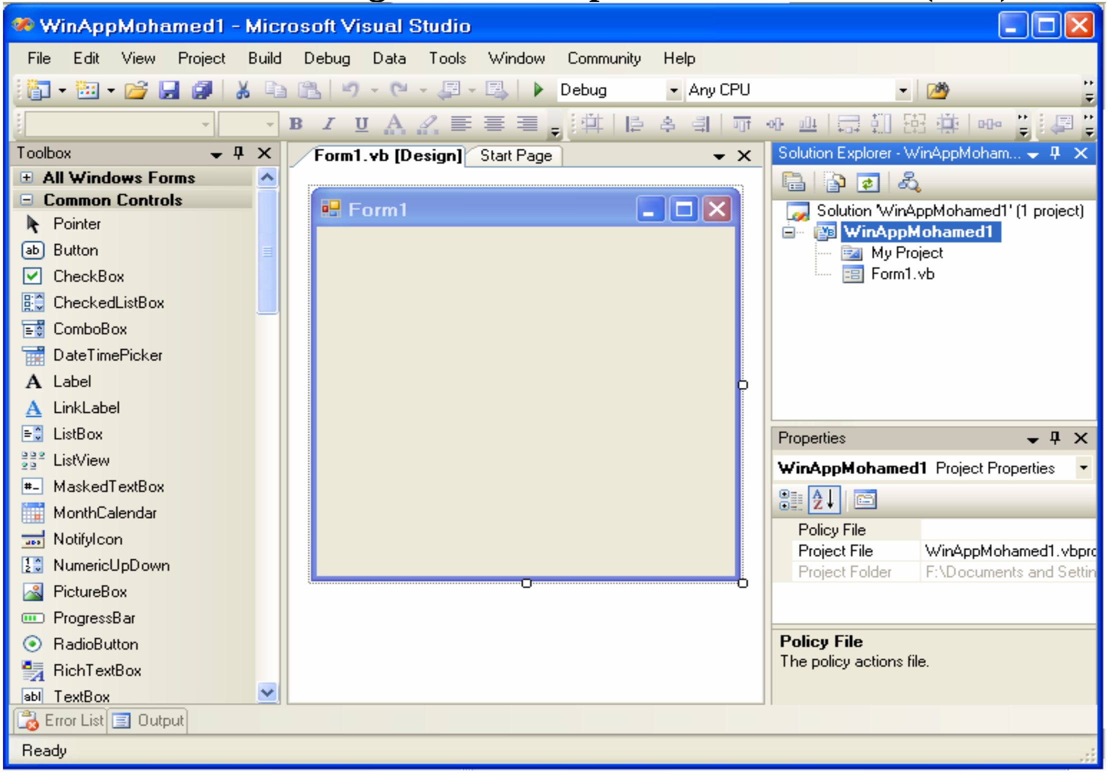
What is Visual Basic .NET?

- **Visual Basic .NET** is part of a grand new version by Microsoft. It is a complete re-engineering of Visual Basic for the Microsoft .NET framework. With Visual Basic .NET, you are able to quickly build Windows-based applications, web-based applications and, eventually, software for other devices, such as palm computers.
- Windows applications built using Visual Basic .NET feature a Graphical User Interface (GUI). Users interact with a set of visual tools (buttons, text boxes, tool bars, and menu items) to make an application do its required tasks.
- Visual Basic .NET Windows applications are **event-driven**, meaning nothing happens until an application is called upon to respond to some event (button pressing, menu selection, ...).

A few of the features of Visual Basic .NET, compared to Visual Basic:

- New Integrated Development Environment
- Uses Object-Oriented Programming (OOP) methods
- New controls and control properties
- Redesigned code window
- Zero-based arrays (no adjustable first dimension)
- Easier to use common dialog boxes
- Structured error-handling (no more On Error Go To)
- New menu design tools
- New techniques for working with sequential files
- All new graphics methods
- New approaches to printing from an application
- Improved support to incorporating help systems in applications
- New web forms for internet applications
- ADO.NET for database access

Visual Basic .NET Integrated Development Environment (IDE)



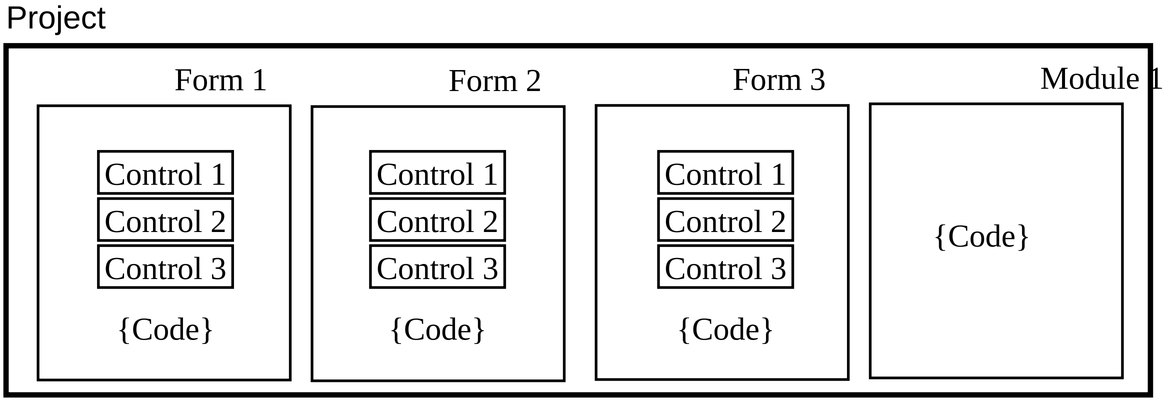
- The **Visual Basic .NET IDE** is where we build and test our application via implementation of the three steps of application development (draw controls, assign properties, write code). As you progress through this course, you will learn how easy-to-use and helpful the IDE is. There are many features in the IDE and many ways to use these features. Here, we will introduce the IDE and some of its features. You must realize, however, that its true utility will become apparent as you use it yourself to build your own applications.

- Several windows appear when you start Visual Basic .NET. Each window can be viewed (made visible or active) by selecting menu options, depressing function keys or using the displayed toolbar.

A *solution* is a container for the projects and solution items that can be built into an application. A solution usually contains one or more related projects.

A *project* is a container within a solution that is used to logically manage, build, and debug the project items that make up your application.

- As mentioned, a project (application) is made up of forms and code.



Application (Project) is made up of:

- **Forms -** Windows that you create for user interface
- **Controls -** Graphical features drawn on forms to allow user interaction (text boxes, labels, scroll bars, buttons, etc.) (Forms and Controls are **objects**.)
- **Properties -** Every characteristic of a form or control is specified by a property. Example properties include names, captions, size, color, position, and contents. Visual Basic .NET applies default properties. You can change properties when designing the application or even when an application is executing.
- **Methods -** Built-in procedures that can be invoked to impart some action to a particular object.
- **Event Procedures - Code** related to some object or control. This is the code that is executed when a certain event occurs. In our applications, this code will be written in the BASIC language (covered in detail in Chapter 2 of these notes).
- **General Procedures - Code** not related to objects. This code must be invoked or called in the application.
- **Modules -** Collection of general procedures, variable declarations, and constant definitions used by an application.

- In a project folder, you will see these files (and possibly more):

AssemblyInfo.vb	Information on how things fit together
SolutionName**.sln**	Solution file for solution named SolutionName
ProjectName**.vbproj**	Project file – one for each project in solution
ProjectName**.vbproj.user**	Another Project file – one for each project in solution
FormName**.resx**	Form resources file – one for each form
FormName**.vb**	Form code file – one for each form
…….**.vb**	Any file contains code like class or form or module has an extension **vb**

Notes

You can write VB.NET code without using Visual Studio .NET or using the IDE. You don't have to install VS.NET to be able to write your code. You can use any text editor to write your code and save the file with .vb extension.

But you have to install the .Net frame work to use the classes located in the class library and to compile your code and to run your application.

The IDE enables Rapid Application Development which means it do most of work for you for example it writes the code for you when you just add a button or any control to the form or project.

Best wishes,

T. Mohamed Ezeddin

For more study materials visit my web site

http://issite.wordpress.com