

Chapter one: Visual Basic 6.0

1.1 Introduction

Visual Basic allows you to create applications that fully exploit the graphical user interface of Windows. The interface of the application is created by arranging objects such as command buttons and text boxes on a form and by writing code to control events. This type of programming is known as Object Orientated Programming (OOP).

1.2 Visual Basic

Visual Basic is a tool that allows you to develop Windows (Graphic User Interface - GUI) applications. The applications have a familiar appearance to the user.

Visual Basic is event-driven, meaning code remains idle until called upon to respond to some event (button pressing, menu selection,). Visual Basic is governed by an event processor. Nothing happens until an event is detected. Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event processor.

VB was created by Microsoft and is a programming language used to develop stand-alone applications. The "Visual" part of the name refers to the method used to create applications and the "Basic" part of the name refers to the BASIC programming language, on which VB is based.

BASIC stands for: *"Beginners All-Purpose Symbolic Instruction Code"*. On the other hand, VBA stands for Visual Basic for Applications and is a scripting language, often used to program in Microsoft Access.

VISUAL BASIC is an Object Oriented Programming (OOP). This is the main divergence from the old BASIC. In BASIC, the program is executed sequentially.

In VISUAL BASIC, due to its Windows Interface, the users may click on a certain object randomly, so each object has to be programmed indepently to be able to response to those actions. Therefore, a VISUAL BASIC Program is made up of may subprograms, each has its own program codes, and each can be executed indepently and at the same time each can be linked together in one way or another.

1.3 Features of Visual Basic

1. Full set of objects - you 'draw' the application.
2. Lots of icons and pictures for your use.
3. Response to mouse and keyboard actions.
4. Clipboard and printer access.
5. Full array of mathematical, string handling and graphics functions.
6. Can handle fixed and dynamic variable and control arrays.
7. Sequential and random access files support.
8. Useful debugger and error-handling facilities.
9. Powerful database access tools.
10. ActiveX support.
11. Package & Deployment Wizard makes distributing your applications simple.

1.4 Visual Basic 6.0 versus Other Versions of Visual Basic

- 1.The original Visual Basic for DOS and Visual Basic For Windows were introduced in 1991.
- 2.Visual Basic 3.0 (a vast improvement over previous versions) was released in 1993.
- 3.Visual Basic 4.0 released in late 1995 (added 32 bit application support).
- 4.Visual Basic 5.0 released in late 1996. New environment, supported creation Of ActiveX controls, deleted 16 bit application support.

And, now Visual Basic 6.0 - some identified new features of Visual Basic 6.0:

- 1-Faster compiler.
- 2-New ActiveX data control object.
- 3-Allows database integration with wide variety of applications.
- 4-New data report designer.
- 5-New Package & Deployment Wizard.
- 6-Additional internet capabilities.

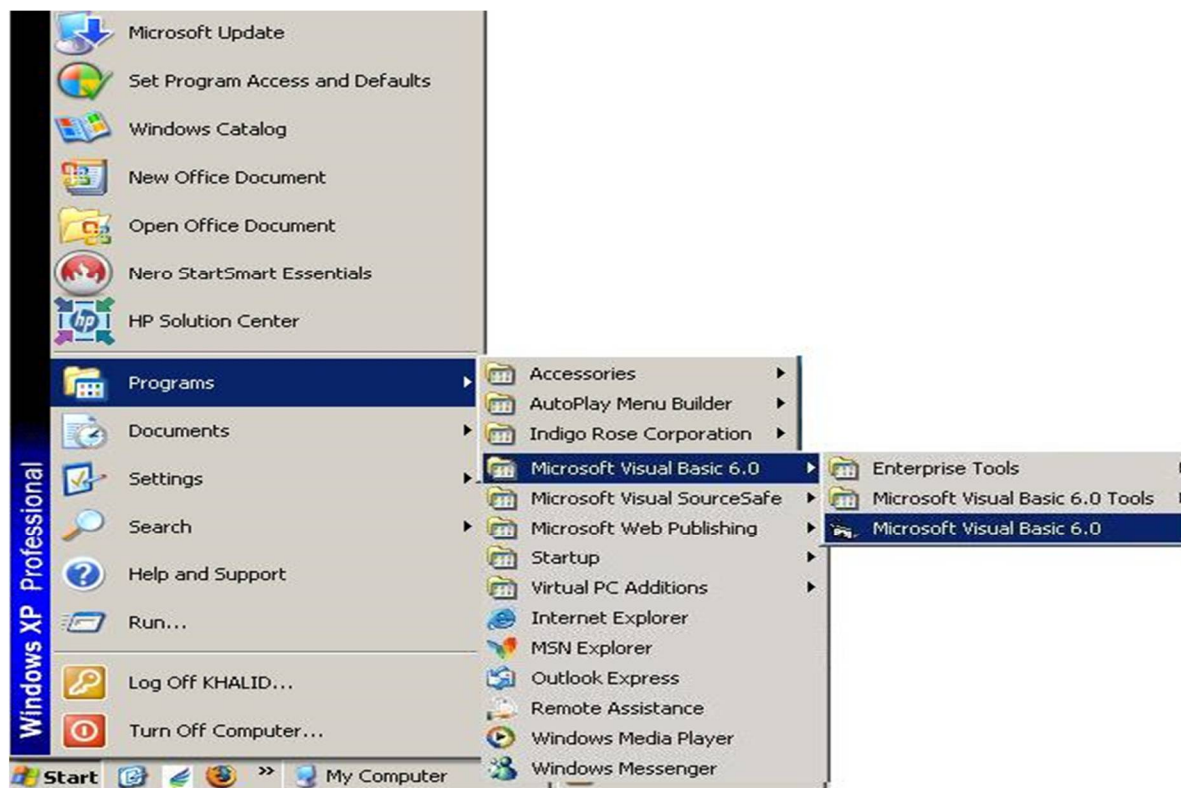
1.5 What programs can you create with Visual Basic 6?

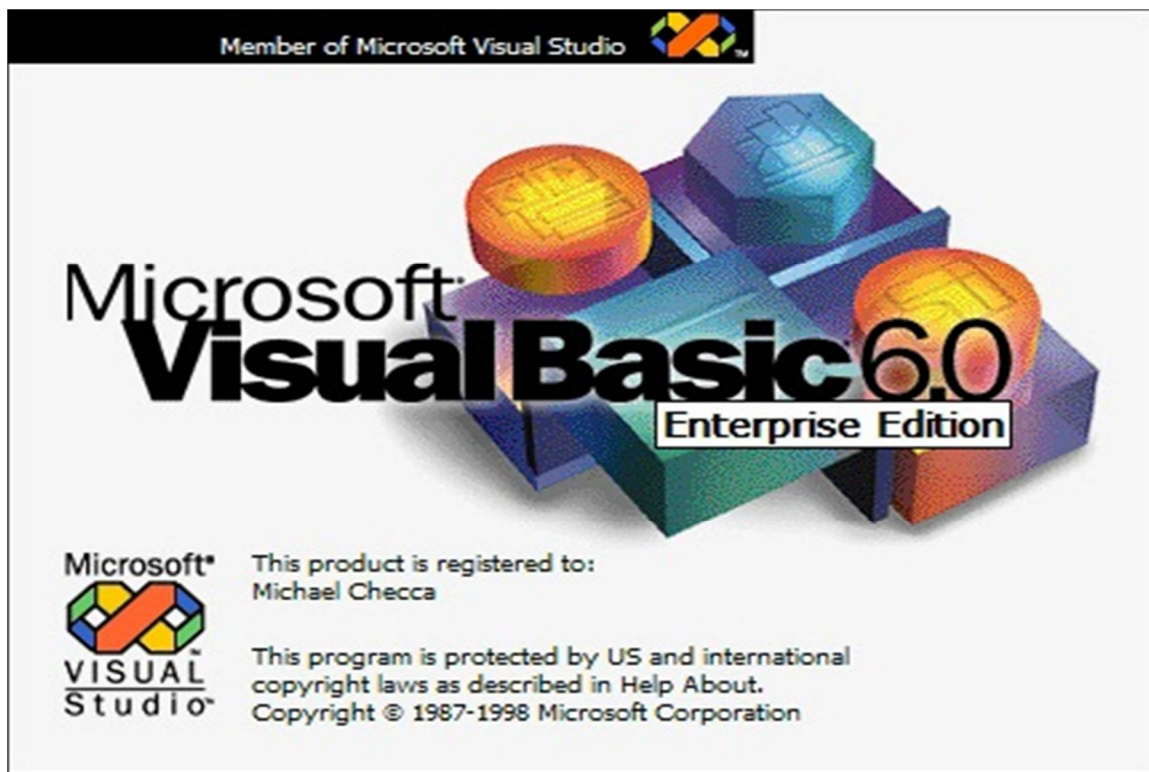
In VB 6, you can create any program depending on your objective. For math teachers, you can create mathematical programs such as Geometric Progression, Quadratic Equation Solver, Simultaneous Equation Solver ,Prime Number Tester, Factors Finder, Quadratic Function Graph Plotter and so on. For science teachers, you can create simulation programs such as Projectile, Simple Harmonic Motion, and Star War etc. If you are in business, you can also

create business applications such as inventory management system, Amortization Calculator , investments calculator, point-of-sale system, payroll system, accounting program and more to help manage your business and increase productivity. For those of you who like games, you can create programs such as slot machine, reverse, tic tac toe and more. Besides, you can create multimedia programs such as Smart Audio Player, Multimedia Player and more. Indeed, there is no limit to what program you can create.

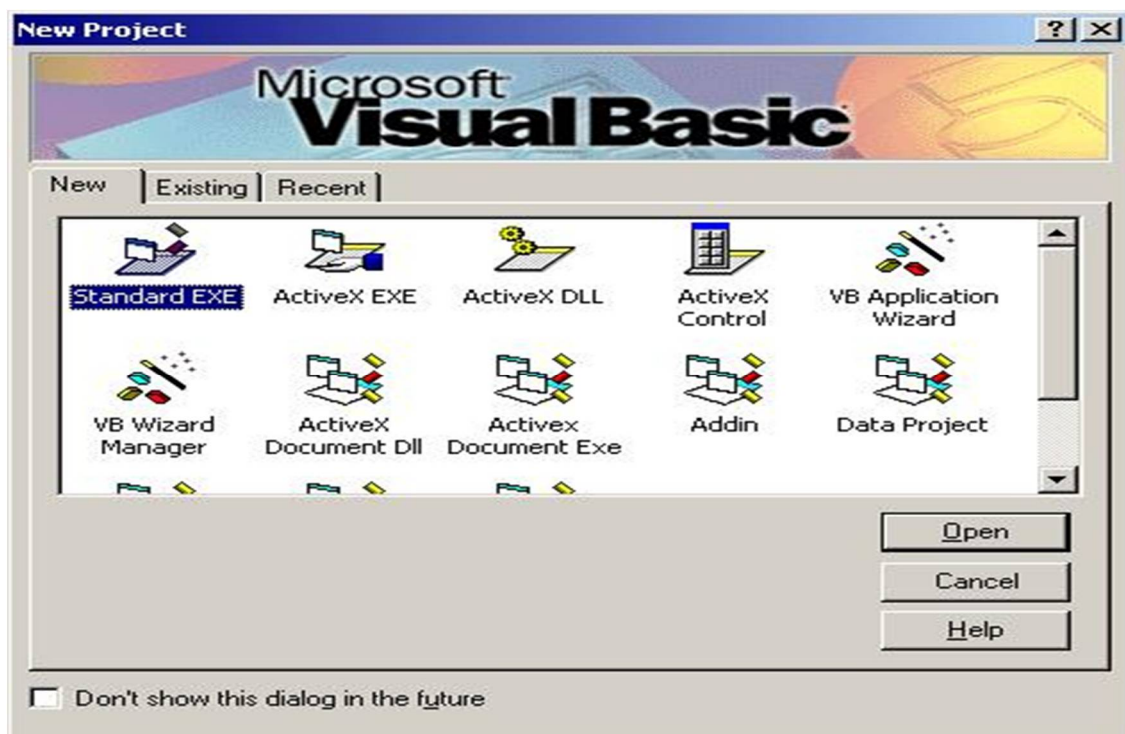
1.6 Starting Visual Basic

To start Visual Basic, choose Visual Basic 6.0 from the start / All Programs / Microsoft Visual Studio 6.0 / Microsoft Visual Basic 6.0. See the following figures.





The first thing you will see is the following screen below, where you are presented with a list of various project templates. Select Standard EXE, and click on Open.



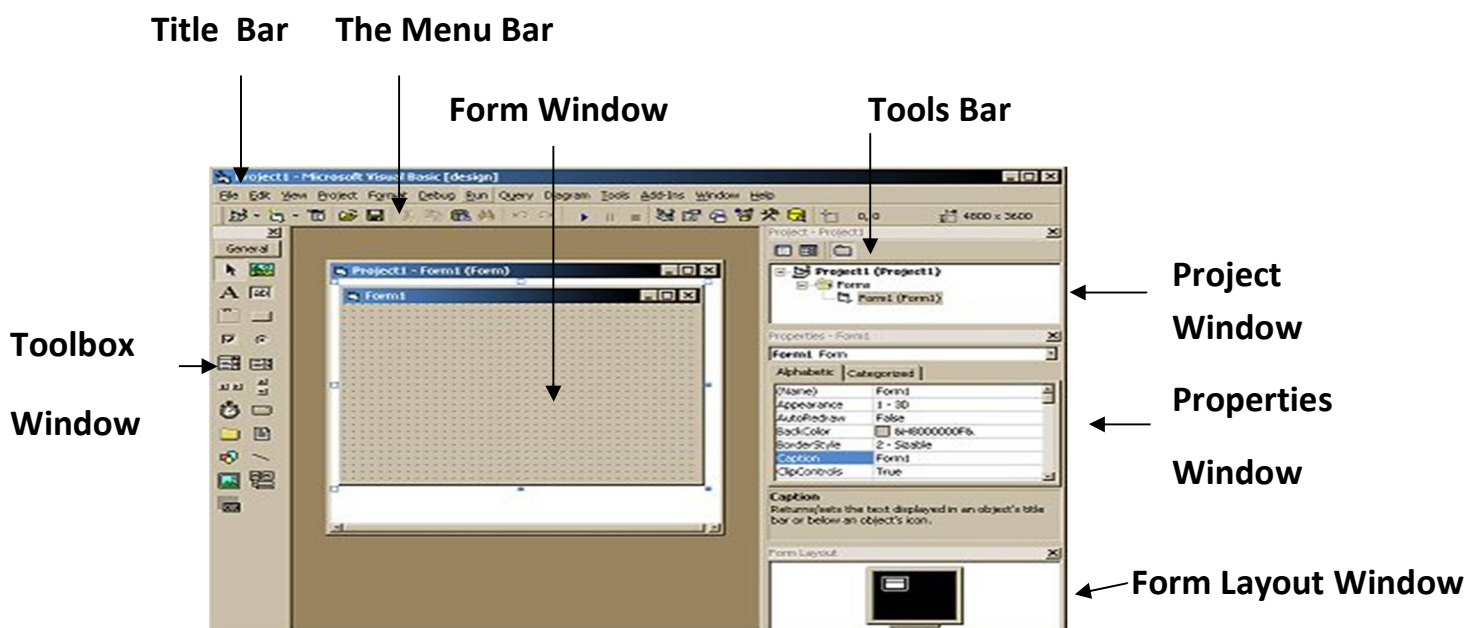
The new project box contains three tabs:

1. **New :-** This tab allows you to choose the type of new project you want to create once you select a specific project type Visual Basic creates the appropriate template for this project. Visual Basic automatically selects the standard type.
2. **Existing: -** This tab allows you to review your existing projects.
3. **Recent: -** Appear the names of the most recent projects you have played, which has been working recently.

Note: - You may not want the dialog box to show a new project at the start of the operation. In this case, we click on the check box (Don't show this dialog in the future) at the bottom of the box. When Visual Basic is started, a default project will start without the dialog box appearing new project.

To re- Show New Project window, choose Tools from the menu bar then Options then the Environment tab and then prompt for project.

And then see the Visual Basic Environment. See the following figure.



Visual Basic Environment:

1-Title Bar

This includes the title of the project.

2. The Menu Bar



Contains thirteen main lists Each list contains a number of commands that the user needs. The Menu contains a few default Windows menu-items (like file, edit, view, help, etc.), and a few functions specific to programming (like Debug, Run, Project, etc.).

- **File-menu:** provides file-related functions. (Open / Save / New / Print)
- **Edit-menu:** provides edit-related functions (Cut / Paste) as well as some look-up functions.
- **View-menu:** provides functions to access various Windows. Project-menu; provides ways to add files, and edit properties of the current project.
- **Format-menu:** provides functions to format controls on the form.
- **Debug-menu:** provides functionality to test your code, as well as tracking down errors.
- **Run-menu:** provides different ways to start and end a program.
- **Query-menu:** provides database-functionality. Only available in Enterprise Edition.
- **Diagram-menu:** provides database-functionality. Only available in Enterprise Edition.
- **Tools-menu:** offers some nifty tools, like wizards. Add-ins-menu; offers ways to integrate various external programs into VB.

- **Window-menu:** provides functionality to rearrange various Windows in the environment.
- **Help-menu:** provides... *drum roll* help.

3. The Tools Bar

the Toolbar provides a number of icons, some of which you will undoubtedly recognize from programs as Word, or any other Windows Application.

Contains four bars for tools:-

- ❖ **Standard Toolbar:** - is the standard toolbar that appears automatically at startup.
- ❖ **Debug Toolbar:** - contains buttons or tools to help track and correct program errors.
- ❖ **Edit Toolbar:** - Contains tools to be used during writing procedure instructions.
- ❖ **Form Toolbar:** - Contains tools that help control the tools in the form.

Note: - To show or hide any of the toolbars, select View and then Toolbars and select the name of the bar from the drop-down list.

We'll briefly describe the various icons here



The first three icons are specific to Visual Basic. The left-most one allows you to add other projects to the project you're working on, the middle icon adds files to your project, and the right-most one allows you to create user-menus.



The next few ought to look rather familiar to anyone who worked with Windows before. The first icon is for Opening a file, the second one for saving a file. The third icon is for cutting, the fourth one for copying, the fifth one for pasting, and the sixth one is for searching. The two arrows on the far right are respectively for undoing and redoing operations.



The first one runs a project, the second one pauses it, and the third one aborts (or stops) the project.

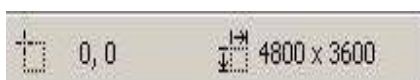


These icons are all short-cuts to the various windows we discussed earlier. The first one is the Project Explorer, the second one the Properties Window, the third one represents the Form Layout Window.

The fourth one is a new one. It points to the Object Browser which we will discuss in future articles.

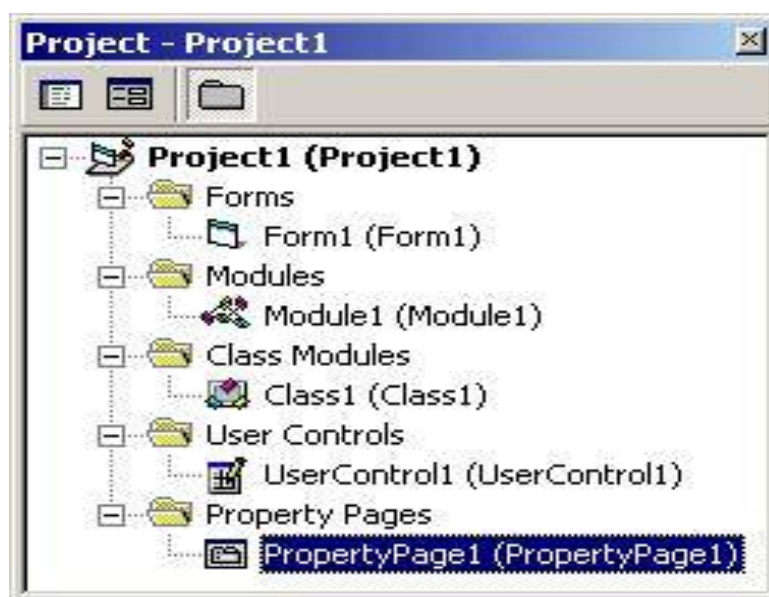
The fifth icon is for the Toolbox, and the sixth is a new one again. It points to the Data View Window, which we will discuss when we get to the article about Connectivity to Databases. (Which will be a few weeks from now).

The right-most icon the Visual Component Manager; basically a repository database for storing reusable bits and pieces of code, files, components, etc. In.



The last part of the toolbar is for keeping track of the position and size of controls. The leftmost part is for the position, while the right part lists the size of a control. To see what both do, double-click on one of the icons on the toolbar, which will make a control be added to your form. Resize and move the control on the form, and the numbers will change.

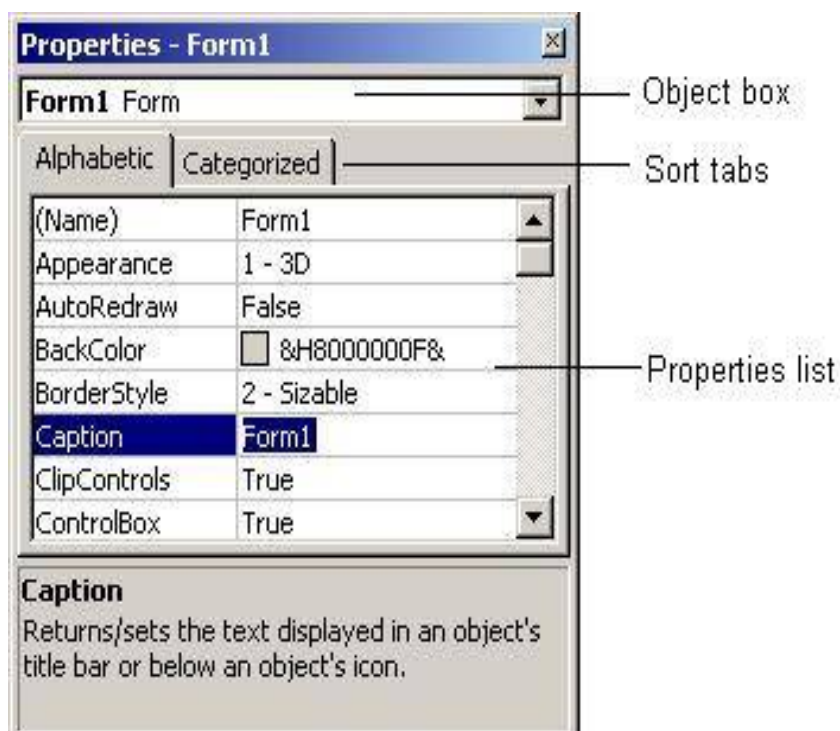
4-The project windows



The Project Window displays a list of all forms and modules making up your application. You can also obtain a view of the Form or Code windows (window containing the actual Basic coding) from the Project window. This window displays projects with all its contents in the form of a tree each project represents the main root of this tree, while the Form files appear in the form of branches of this tree. The Project Explorer window contains three buttons:

- ❖ **Toggle Folders:** Clicking this button changes the objects view in the window.
- ❖ **View Object:** Clicking on this button shows the visual interface of the form and all elements Control.
- ❖ **View Code:** Clicking this button shows the code of the form and the code of all the controls set up.

5- The properties window



It shows the properties of the currently selected object. The definition of a properties is: *"A property is a characteristic of an object, such as size, caption, or color"*.. You can Visual Basic applies default properties change properties at design time or run time.

To set the properties of an object, choose Properties from the Window menu click the Properties button on the toolbar or press F4

The Properties Window is made up out of three elements (As seen in the picture):

1-Object Box

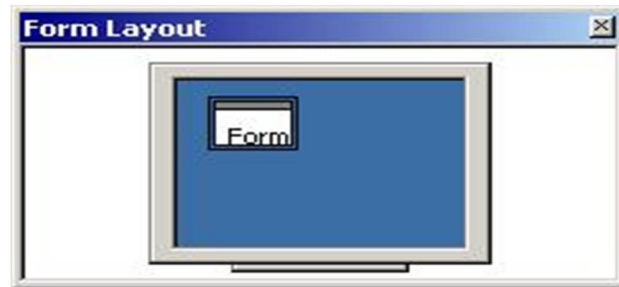
2- Sort Tabs

3- Property List

- ✚ **The Object Box:** provides a list of objects currently in use in the project, of which you can set properties.
- ✚ **The Sort Tabs:** provide a quick way of sorting the Property List. You can sort either by category, or alphabetically. This is really a matter of choice, or habit. Use whichever way you prefer.
- ✚ **The Property List:** provides a list of all the properties that can be set for the currently selected object. The left column provides the property name; the right column shows the currently assigned value.

Under the Property List, a short description of the currently selected property is displayed. (You can turn this off by right-clicking the Property Window, and unchecking the Description option).

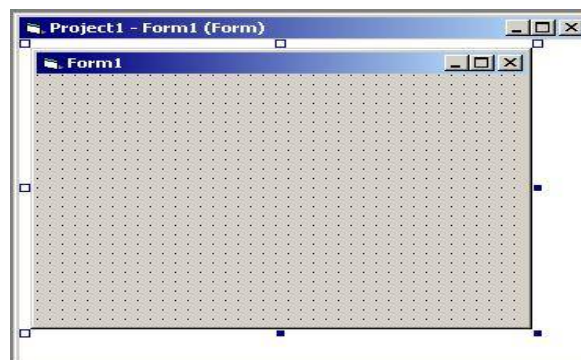
6. The Form Layout Window



Using the Form Layout Window, you can choose where your program will essentially show up once it is completed. As you can see, it already contains a small version of a form, which can be moved around the screen.

Right-clicking the Form Layout Window will also give you the possibility to select some positions (In the 'Startup Position'-sub menu). The Form Layout Window is not.

7-The Form Window



Within the main application window is the form window with a title bar containing the default name Form1. In this window, you can use the toolbox and the toolbar to add controls, graphics and pictures to create the look of your application.

8. The Toolbox Window



The toolbox holds various ActiveX controls that you can use in your projects. In addition to the standard ones you already see, you can also add other controls (that could be provided by Microsoft, third-parties, or controls that you made yourself).

Now we will explain the standard toolbox



Picture Box

This control is like a form within a form. It's primary use is to display pictures, but it can also be used to contain other objects within a group like a frame, or to display messages. The controls can simply be dragged into the Picture box while in design time to make them part of picture box. It is a good tool for when you want a colored region of a form or when you would like to group controls on the form separately.



Command button

Creates a button the user can choose to carry out a command. Examples include submitting information, cancelling a process, or exiting a program.



Labels

Labels are used to display static text in an application. With static, I mean that while the application is running, the text cannot be altered.



Text Box

Holds text that the user can either enter or change.

Combo Box

Allows you to draw a combination list box and text box. The user can either choose an item from the list or enter a value in the text box.

Frames



Frames are used to group certain controls together graphical or functional grouping. To group controls, draw the Frame first, and then draw controls inside the frame.



Drive List Box

Displays valid disk drives



Dir List Box (directory list box)

Displays directories and paths.



List Box

Used to display a list of items from which the user can choose one. The list can be scrolled if it has more items than can be displayed at one time.

Displays a list of files.



Line

Used to draw a variety of line styles on your form at design time.



Shape

Allows you to draw a variety of shapes on your form at design time. You can choose a rectangle, rounded rectangle, square, rounded square, oval, or circle.



Image

Displays a graphical image from a bitmap, icon, or metafile on your form.
Images



Timer

Generates timer events at set intervals. This control is invisible at run time.

HScroll Bar



Provides a graphical tool for quickly navigating through a long list of items or a large amount of information, for indicating current position.



VScroll Bar

Provides a graphical tool for quickly navigating through a long list of items or a large amount of information, for indicating current position.



Data

Provides access to data in databases through bound controls on your form.



Pointer

The only item in the Toolbox that doesn't draw a control. When you select the pointer, you can only resize or move a control that has already been drawn on a form.



Check Box

Multiple checkboxes together (For instance grouped in a frame), every single Check Box can be either checked or unchecked A Check Box has two values. Checked, or unchecked.



Option Button

An Option Button is similar to a Check Box in that it can be either 'checked' (With an Option Button, this is called 'True') or 'unchecked' (False). However, if grouped together, only ONE of the Option Buttons can be true. The Option Button that previously was true will be false as soon as a new Option Button is set to True.



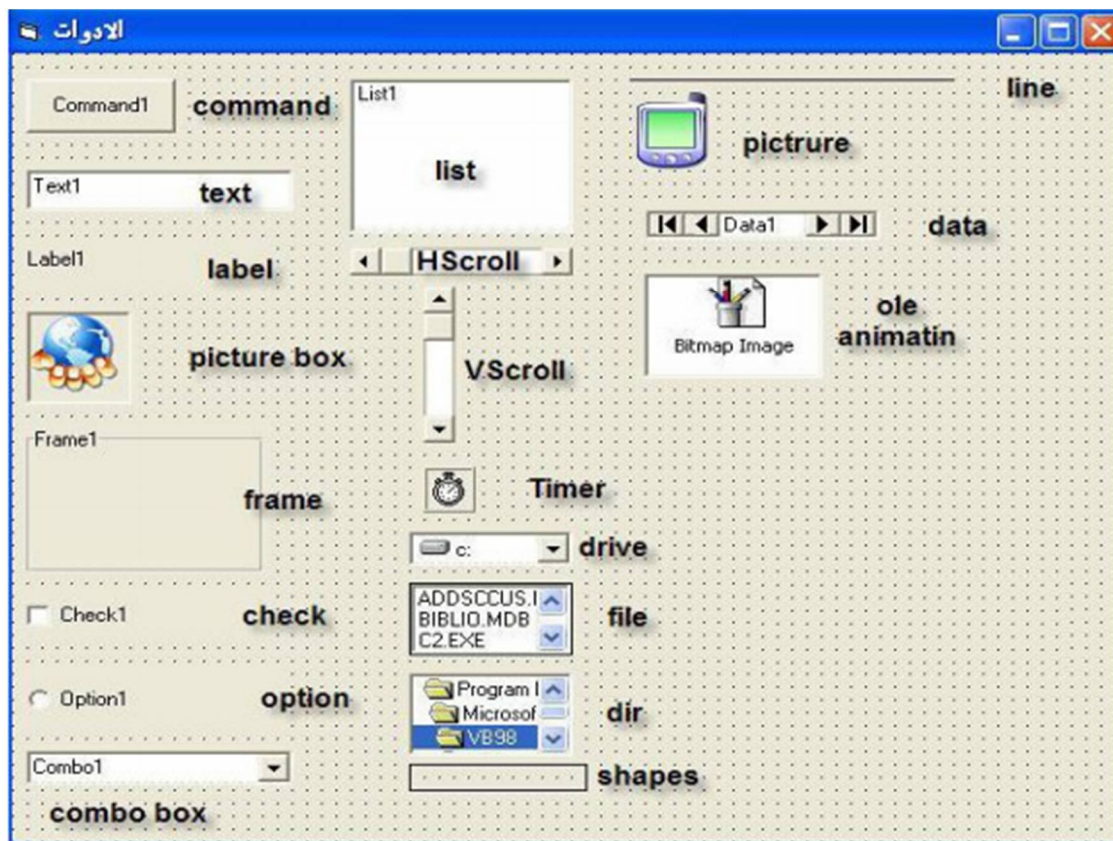
File list box

Displays a list of files

OLE

OLE

Allows you to link and embed objects from other applications in your Visual Basic application.



Form the tools on the model

9. Code Window

Used to write the program by defining variables and defining user subprograms. The first line of the code begins with **Private sub** followed by the name of the event or tool and then the event under which that element or tool will operate. The last line of the code is **End sub** and cannot be without any of these two sentences at all.

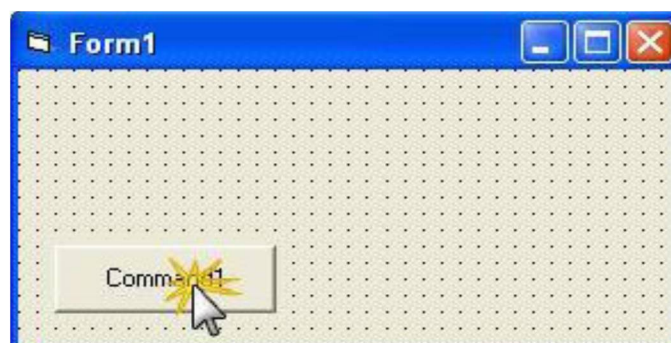


We can show the code window in one of two ways:

1. Through the Project Explorer window by clicking on the button to display the code window.
2. Double-click by the left mouse button on the object to display its code.

How to write code: -

Double-click the item that we want to write its code, see the following figure.



The code window appears, see the following figure.

