# 17. Introduction to Visual Basic Programming

Visual Basic (VB) is the fastest and easiest way to create applications for MS Windows. Whether you are an experienced professional or brand new to Windows programming, VB provides you with a complete set of tools to simplify rapid application development.

## 17.1 What is Visual Basic?

The “Visual” part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply put prebuilt objects into place on screen.

The “Basic” part refers to the BASIC (Beginners All-purpose Symbolic Instruction Code) language. We need to notice that VB has evolved from original BASIC language and now contains several hundred statements, functions, and keywords, many of which relate directly to the Windows GUI.

The VB programming is not unique to Visual Basic. The following applications or programming also used VB:

- VBA in MS Word, Excel, and Access
- VB Script for web-based programming

### (1) Visual Basic Concepts

It is necessary for us to have a better understanding some of the key concepts VB has built. A simplified version of the workings of Windows involves three key concepts – windows, events, and messages.

A window is a rectangular region with its own boundaries. You are probably already aware of several different types of windows: an explorer window, a document window, or dialog box. Other types of windows include:

- command buttons
- icons
- text boxes
- option buttons, and
- menu bars

The MS Windows operating system manages all of these many windows by assigning each one a unique id number (window handle or hWnd). Events can occur through user actions such as mouse click or a key press, through programmatic control, or even as a result of another window’s actions. Each time an event occurs, it causes a message to be sent to the operating system. The system processes the message and broadcasts it to the other windows.

(2) Event-driven programming vs. Procedural programming

In traditional or procedural applications, the application itself controls which portions of code execute and in what sequence. Execution starts with the first line of code and follows a predefined path through the application, calling procedures as needed.

In an event-driven application, the code does not follow a predetermined path – it executes different code sections in response to events. Events can be triggered by:

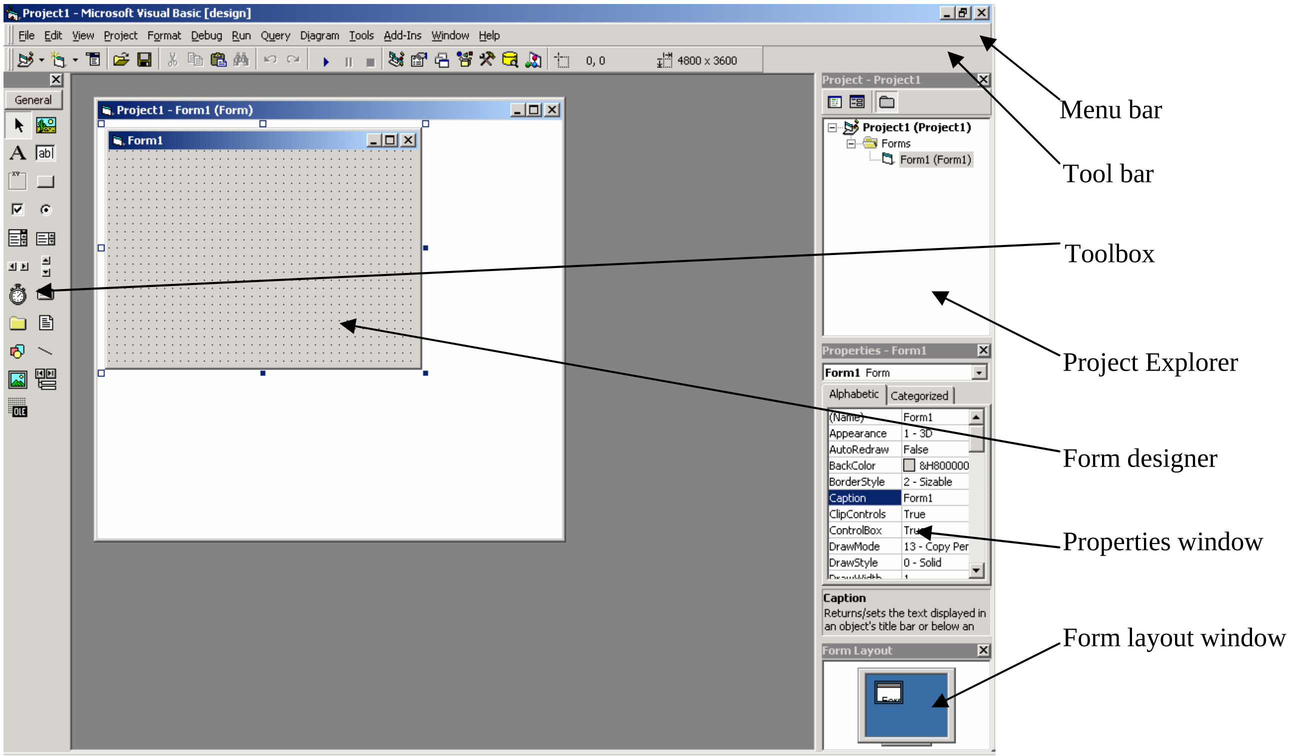
- the user's actions,
- messages from the system or other applications, or
- event from the application itself.

(3) VB Integrated Development Environment

VB Integrated Development Environment (IDE) integrates many different functions such as design, editing, compiling, and debugging within a common environment while each of these tools would operate as a separate program in a traditional development environment.

Starting VB IDE

You can start VB IDE either from clicking Start on the Taskbar or clicking VB icon on the desktop (Figure 1).



| | |
|---|---|
| (Name) | Form1 |
| Appearance | 1 - 3D |
| AutoRedraw | False |
| BackColor | &H800000 |
| BorderStyle | 2 - Sizable |
| Caption | Form1 |
| ClipControls | True |
| ControlBox | True |
| DrawMode | 13 - Copy Per |
| DrawStyle | 0 - Solid |

**Caption**
Returns/sets the text displayed in an object's title bar or below an

Figure 1. VB IDE.

- Object browser
- Code editor window
- Form layout window – allows you to position the forms in your application using a small graphical representation of the screen.
- Immediate, Locals, and Watch windows under View menu are used for debugging.

Environment Options

VB provides a great deal of flexibility, allowing you to configure the working environment to best suit your individual style. Two different styles are available for the VB IDE:

- single document interface (SDI): with SDI, all of the IDE windows are free to be moved anywhere on screen.
- Multiple document interface (MDI): all of the IDE windows are contained within a single resizable parent window.

To switch between SDI and MDI modes,

a. Select Options from the Tools menu.
   The Options dialog box is displayed.

b. Select the Advanced tab.

c. Check or uncheck the SDI Development Environment check box.

The IDE will start in the selected mode the next time you start Visual Basic.

## 17.2 VB Programming Examples

(1) Your First VB Application

I am using the classical example of “Hello World!” in programming textbook. There are three main steps to creating an application in Visual Basic:

- Create the interface
- Set properties
- Write code.

In our example here, we want to display “Hello World!” in a text box once you click a command button.

- Creating interface: add a text box and a command button

- Setting properties

- Writing code: double click the command button, then code editor window will be displayed, type text1.text = "Hello World!"

- Running application

(2) Simple VB Example for Calculating Basal Area of a Tree

Once the user enters the DBH for a tree, the basal area for that tree should be displayed in a text box. The basal area is calculated by using the following equation:

$$BA = 0.005454154*DBH^2$$

Where, BA is basal area in $ft^2$ and DBH is the tree's diameter at breast height in inches.

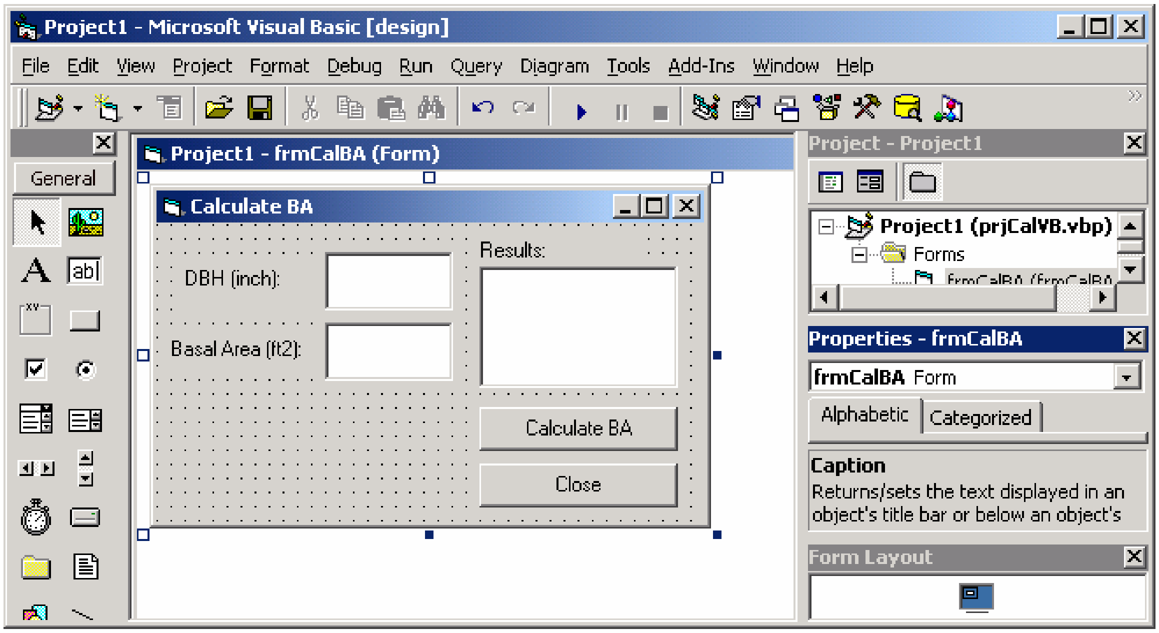
The results should be displayed in a list box for comparison among trees. Here are the controls we need in this project:

- text box
- list box
- command button

Creating a VB Project

From the Start menu, click All Programs|MS Visual Studio 6.0|Visual Basic 6.0. You start a new project by choosing New Project from the File menu, then selecting Standard EXE in the New Project dialog box (when you first start Visual Basic, the New Project dialog box is presented). VB creates a new project and displays a new form for you. Now we need to design the interface. What you need to do are as follows:

- Create a directory like that C:\For240\VBApps\CalBA\
- Start VB
- Put VB controls on the form (Figure 2)
- Name the controls (Table 1)
- Clear the default texts in the boxes
- Change the form caption to Calculate BA
- Save the project as prjCalBA



Caption
Returns/sets the text displayed in an object's title bar or below an object's

Figure 2. Interface for calculating BA.

Table 1. Property settings of the objects.

| Object | Property | Setting |
|---|---|---|
| Form | Name<br>Caption | frmCalBA<br>Calculate BA |
| Text1 | Name<br>Text | txtDBH |
| Text2 | Name<br>Text | txtBA |
| List1 | Name<br>List | lstResult |
| Command1 | Name<br>Caption | cmdCalBA<br>Calculate BA |
| Command2 | Name<br>Caption | cmdClose<br>Close |

<u>Coding</u>

Double click the command button and a code-editing box will pop out. Type the following lines under the command button 1 (cmdCalBA).

```
Private Sub cmdCalBA_Click()

   Dim DBH, BA

   DBH = txtDBH.Text
   BA = 0.005454154 * DBH * DBH
   txtBA.Text = BA

   lstResult.AddItem DBH & ", " & BA

End Sub
```

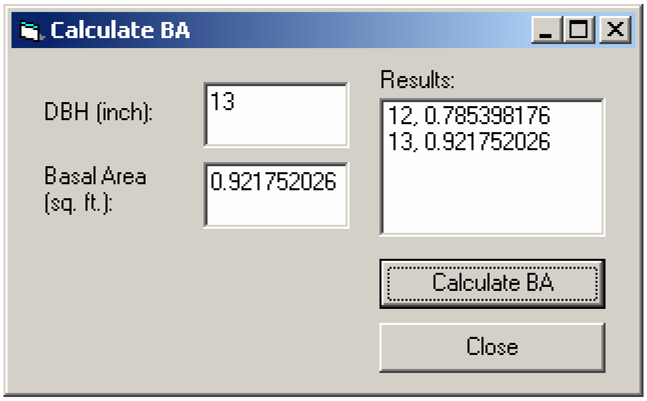
Under command button 2 (cmdClose), type:

```
Private Sub cmdClose_Click()

   End

End Sub
```

Remember!! Now, you need to save the project again by clicking the save button on the menu bar.

### Run the Project

Use the arrow button on the menu bar to run the project. You enter 12 in DBH box, then click ‘Calculate BA’ button, you will add the first result to the list box. If you change the DBH from 12 to 13, then click the ‘Calculate BA’ button; you will add the second result to the box (Figure 3). You can repeat the above procedures as you wish.



Calculate BA

DBH (inch): 13

Basal Area (sq. ft.): 0.921752026

Results:

```
12, 0.785398176
13, 0.921752026
```

Figure 3. Output of BA.

(3) Database Connection/Display Application

This application demonstrates how a data control and a DB grid control can be used to display a table of information from a database. VB makes it easy to access database information from within your application. The data control provides the ability to navigate through the database recordset, synchronizing the display of records in the grid control with the position in the recordset.

The following controls are needed in this application:

- Data control
- DBGrid control
- Command buttons

The database we are going to use is dbStudent created in the database application section. The recordset is table tblCourse in the database.

Creating a Project

You begin creating the application by choosing New Project from the File menu, then selecting Standard EXE in the New Project dialog box (when you first start Visual Basic, the New Project dialog box is presented). VB creates a new project and displays a new form. Now we need to design the interface – putting data control, DBGrid control and buttons on the form. Since the DBGrid is not in the default toolbox, we need to add it there. What we can do are:

a. Select Components under Project menu, then the Components dialog box will be displayed.
b. Find Microsoft Data Bound Grid Control 5.0 (SP3) in the controls list box and check the box to its left.
c. Click the OK button, the icon for the DBGrid control will appear in the toolbox.

Use the toolbox to draw the controls on the form (Figure 4).

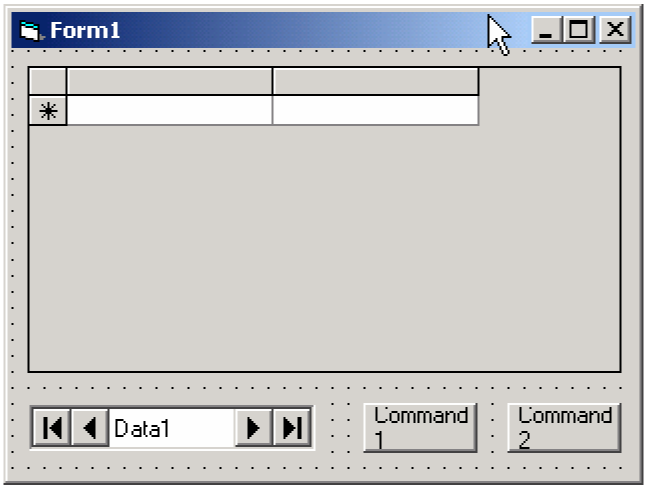


Figure 4. Controls on the form.

### Setting Properties

In the Properties window, set properties for the objects according to Table 2. Use the default settings for all other properties.

Table 2. Property settings of the objects.

| Object | Property | Setting |
|---|---|---|
| Form | Name | frmCourse |
| | Caption | Firstapp |
| Data1 | Name | dataCourse |
| | Caption | Courses |
| DBGrid1 | Name | DBGridCourse |
| | Caption | Courses |
| | DataSource | dataCourse |
| Command1 | Name | cmdDelete |
| | Caption | Delete |
| Command2 | Name | cmdClose |
| | Caption | Close |

Now, save your project with a name of prjFirstapp. The interface of your project will look like that (Figure 5).

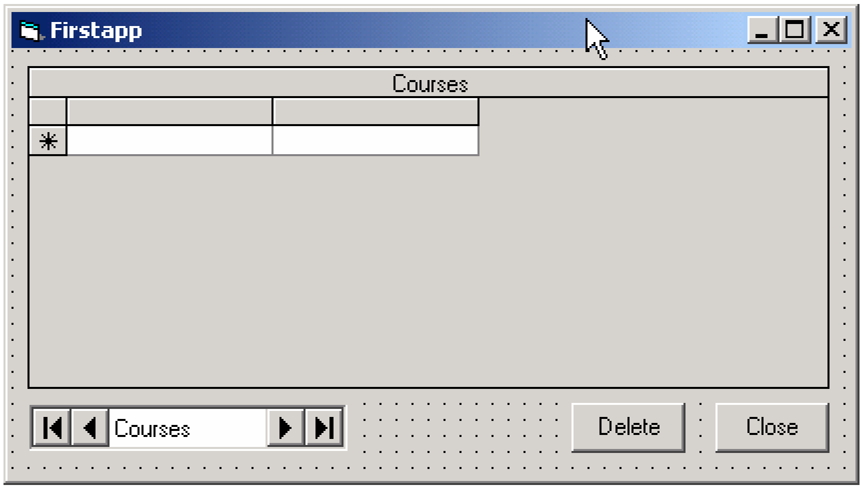


Figure 5. The interface of your application.

### Writing Event Code

Double-click the form or control to display the Code window, and then type the code for each event procedure.

Add this code to Form_Load event procedure to connect the database and retrieve data from the table when the program first starts.

```
Private Sub Form_Load()

   ChDir App.Path  'Change to application directory
   dataCourse.DatabaseName = "dbStudent.mdb"   'set database name
   dataCourse.RecordSource = "Select * from tblCourse" 'set recoredset
   dataCourse.Visible = False  'make data control invisible

End Sub
```

Add the following code to cmdDelete_Click event procedure to delete a selected record from the DBGrid when the Delete button is clicked.

```
Private Sub cmdDelete_Click()

   dataCourse.Recordset.Delete 'delete a selected record from the table

End Sub
```

Add the code to cmdClose_Click event procedure to end the application when you click Close button.

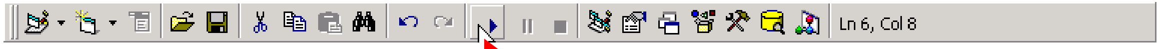
```
Private Sub cmdClose_Click()

   Unload Me   'unload the form
   End 'end the application

End Sub
```

Again, save your project at this point.

## Running the Project

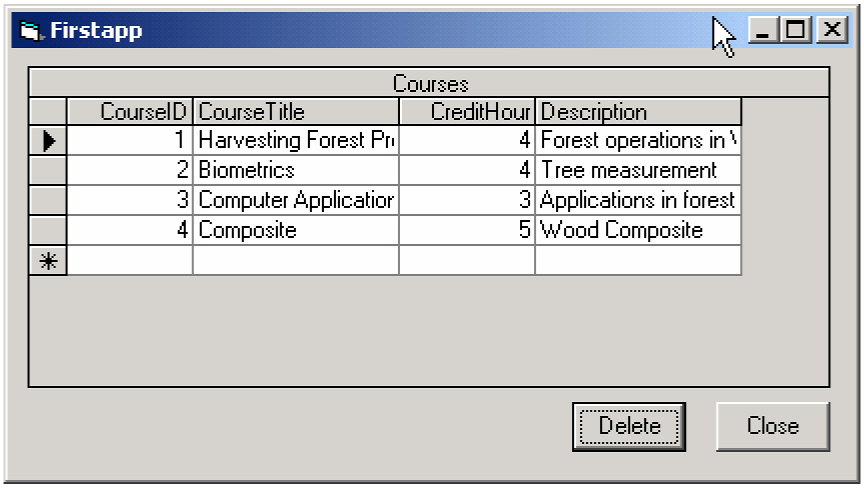
There are two ways that you can use to run the application:

a. From Run menu, click Start
b. From toolbar, click the button



Start

Your application will be in the running mode (Figure 6).



Courses

| | CourseID | CourseTitle | CreditHour | Description |
|---|---|---|---|---|
| ▶ | 1 | Harvesting Forest Pr | 4 | Forest operations in |
| | 2 | Biometrics | 4 | Tree measurement |
| | 3 | Computer Applicatior | 3 | Applications in forest |
| | 4 | Composite | 5 | Wood Composite |
| * | | | | |

Figure 6. Running the application.

You may need to test your program:
a. Add new courses
b. Edit existed courses
c. Delete a course
d. Close your application

## References

Microsoft Corporation. 1998. Visual Basic 6.0 – Programmer's guide. Microsoft Press. Redmond, WA.