

The Events:

Events are set of procedures that the user must understand to write programs in (Visual Basic.6), the events help the programmer to design the program and control its parts.

Events are handled by keyboard input, mouse input and output...etc, in (Visual Basic.6). Creating events starts by selecting the desired object that the user wishes to accomplish with in his program. The left combo box which is in the code window must be pressed. A dropdown list will appear to select the desired object that the user needs to use in his program as seen in figure (1) below:

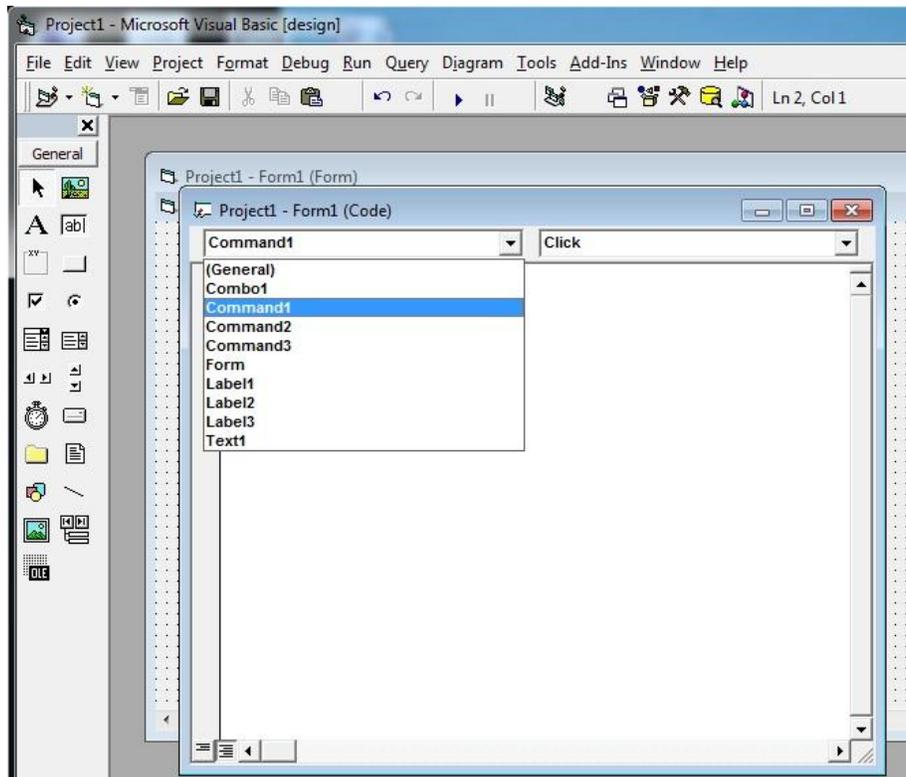


Figure (1).

The second step is pressing on the right combo box which is also on the code window. Here the user must select the desired event from the dropdown list in the combo box to use in his program as seen in figure (2) below:

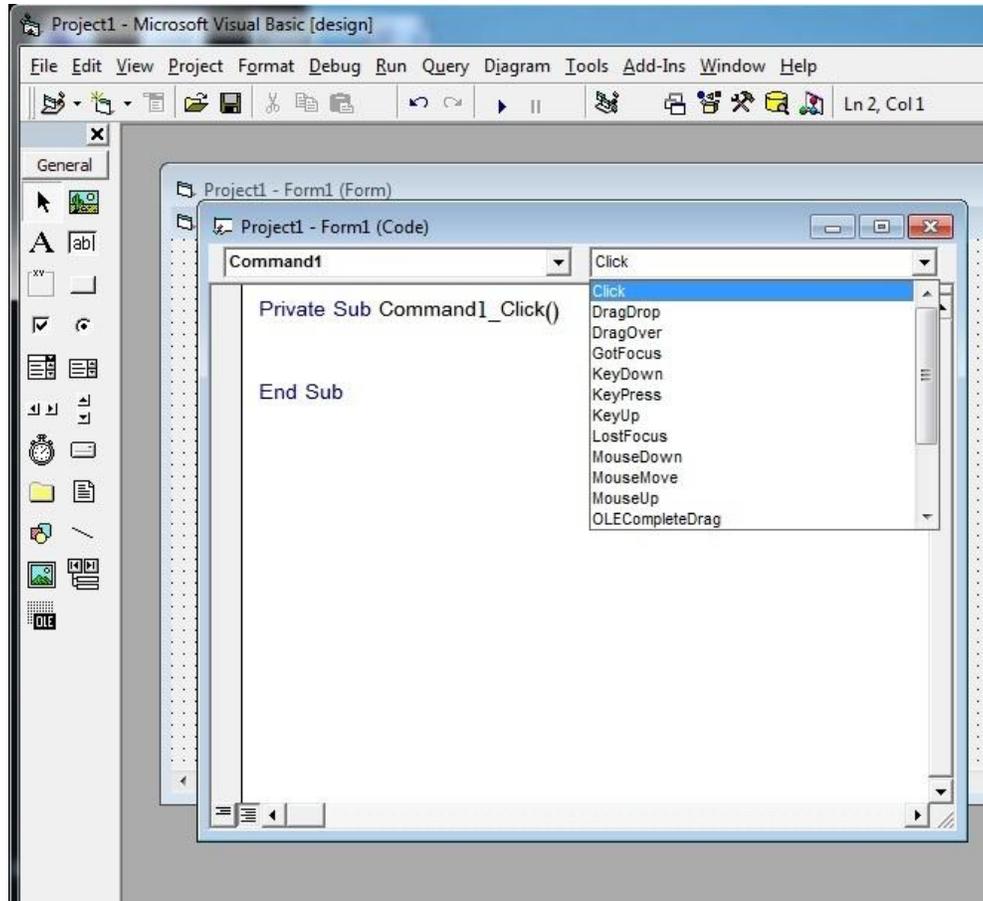


Figure (2).

The final step is to write the codes that controls that object. Events can be classified into two categories:

1- Form Events:

Visual Basic forms and controls can trigger dozens of events in your applications. Forms, controls, and classes all have events. Below are some of the events which might be used by the user:

- Load:
Load is the event most often used to initialize any dynamic components of

the form, such as private data, control arrays, or any other element of the design that needs to be setup at run time rather than in design view.

- Activate:

This event occurs whenever the form window gets the input focus. If you have code that tracks the active form in the application, you might use this event for that code.

- Paint:

This is where VB actually draw the form on the screen. If you are drawing directly on the form with graphics methods, that code might go in this event.

2- Control Events:

Controls also have events, but the events which occur vary by the type of control. Most controls will have a full set of keyboard and mouse events, including (KeyUp, KeyDown, KeyPress, MouseUp, MouseDown, MouseMove, Click, and DbClick which is mean double click). The nature of the events is exactly what you would expect given the name of the event. There are also events to support drag and drop operations, including (DragDrop and DragOver).

Some other common control events are Change, which occurs when the data in a control is changed like (GotFocus and LostFocus), which occur when the user enters or leaves the control using the mouse or keyboard.

Among all the control events, the one which is used most often in applications is the click event - normally attached to command button controls and menu controls.

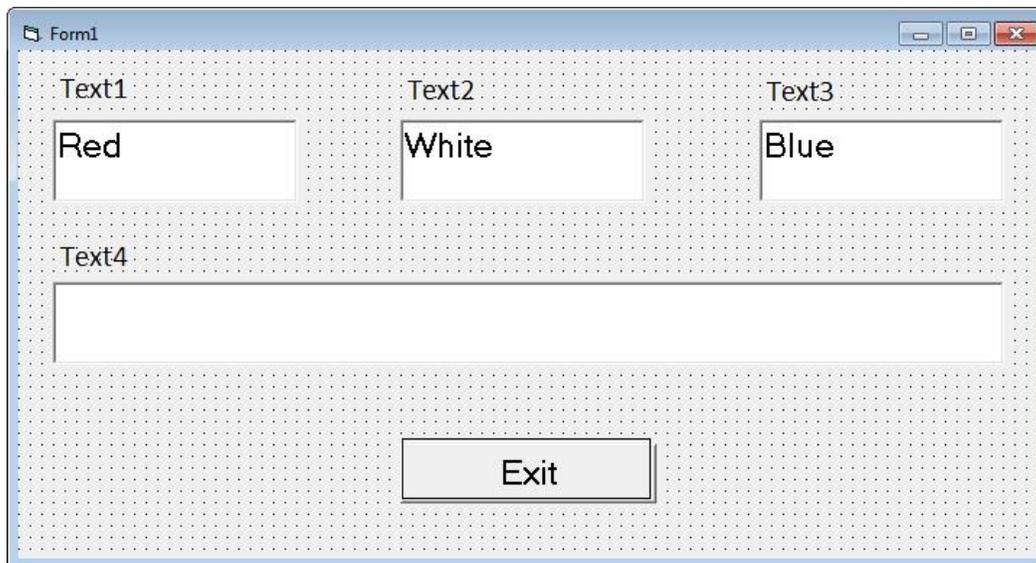
For more understanding see the example below:

Ex: Design a program that is able to print on a (Text box) object the script (Hi: This is the computer, move the mouse over the objects please.). Create three (Text box) objects and write the word (Red) in the first (Text box) object, (White) in the second and (Blue) in the last (Text box) object. Change the color of the form to (Red, White or Blue) as the user move the mouse above these (Text boxes). Create a (Command button) object to exit from the program. If the user move the mouse

in any location except these (Text boxes) let the program change the color of the form to (Green).

So:

1- The design part: The programmer can design the program as below:



2- The coding part: the programmer must write the codes as below:

```
Private Sub Command1_Click( )
```

```
End
```

```
End Sub
```

```
Private Sub Form_Load( )
```

```
Text4.Text = "Hi: This is the computer, move the mouse over the objects please."
```

```
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Form1.BackColor = vbGreen
```

```
End Sub
```

```
Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Form1.BackColor = vbRed
```

```
End Sub
```

```
Private Sub Text2_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Form1.BackColor = vbWhite
```

```
End Sub
```

```
Private Sub Text3_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Form1.BackColor = vbBlue
```

```
End Sub
```